Emergent Engineering Design:
Design Creativity and Optimality Inspired by Nature

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Rafal P. Kicinger
Master of Science, Civil Engineering
Warsaw University of Technology, 1999

Director: Tomasz Arciszewski, Professor
Department of Civil, Environmental and Infrastructure Engineering

Fall Semester 2004
George Mason University
Fairfax, Virginia

DEDICATION

To my wife Iwona
In your eyes and heart I discovered the most important answers

ACKNOWLEDGEMENTS

I would like to thank my dissertation committee members: Dr. Arciszewski, Dr. De Jong, Dr. Rehder, Dr. Sauer and Dr. Schum. In particular, I want to thank my advisor, Dr. Arciszewski, for his time, inspiration, help and guidance, and for providing excellent and intellectually challenging research environment throughout my four years at George Mason University. Without him, this intellectual endeavor would be simply impossible. I would also like to thank Dr. Sauer. Nearly all I know about dynamical systems and chaos theory I have learned from his classes and our 'chaotic Friday' meetings. Likewise, I thank Dr. De Jong and fellow members of the Genetic Algorithms Group. Thanks to them, I first became aware of, then became interested in, and finally became inspired by evolutionary algorithms.

I also acknowledge continuous support from George Mason University. The Interdisciplinary Scholarships from the Office of the Provost and the Graduate Research Assistantships from the School of Information Technology and Engineering provided essential financial support which helped me focusing on fundamental research and study and made this dissertation possible.

I would like to thank the staff of the Department of Civil, Environmental and Infrastructure Engineering and the Office of International Programs and Services at George Mason University. They have always been very supportive and made the entire process of my study at GMU much easier.

The next group of acknowledgements goes to the IT&E Computing Resources and specifically to Jonathan Goldman and Larry Jackson. They supplied me with the computational resources I could only dream of (almost 50 computers working for many months which generated more than 700 GB of experimental data) and greatly contributed to the success of the empirical part of this dissertation. Likewise, I would like to thank a group of people from Wolfram Research, Inc. They offered helpful suggestions and inspiring discussions during the 1st NKS Summer School I attended. They also provided me with all necessary licenses of Mathematica free of charge for this dissertation project. I'm also indebted to their support team which helped with the nitty-gritty of the Mathematica programming language and its integration with Java.

My intellectual adventure in this country would not even begin without the help of Dr. Gizejowski, at that time the Dean of the Department of Civil Engineering at Warsaw University of Technology. I would like to thank him for opening the door of science to me and for his constant support and encouragement.

Besides, I owe a lot to my parents. They have been with me in all the paths of my life and have always provided encouragement and wholehearted support. And last, but definitely not least, I would like to thank my wife for her love, smile, and support. No words can express how much she contributed to the success of this endeavor and to my personal and intellectual growth.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

EMERGENT ENGINEERING DESIGN: DESIGN CREATIVITY AND OPTIMALITY INSPIRED BY NATURE

Rafal P. Kicinger
George Mason University, 2004
Dissertation Director: Dr. Tomasz Arciszewski

For a long time, engineering design research has been focused on the development of various design theories, methodologies, methods, tools, and procedures. The design methods have been subsequently used by engineers to more efficiently design artifacts. However, as the artifacts have grown in complexity, the need for new methods has become obvious. Also, in a nowadays world, increased competition and globalization require organizations to reexamine traditional product development strategies. Traditional methods focused exclusively on the numerical optimality of produced artifacts, or their manufacturing processes, are no longer adequate. Creativity and innovation of designed artifacts provide organizations not only with a competitive advantage but are, in fact, a matter of their survival.

This dissertation addresses this problem by posing and answering the question: "How can one construct an effective method for designing engineering systems that would support development of novel/creative designs and their efficient optimization?" It proposes a new and conceptually coherent design method, called Emergent Engineering Design. The proposed design method is inspired by the fundamental processes occurring in nature, which has arguably created the most fascinating designs known to humankind. All major phases of Emergent Engineering Design are represented by complex systems, including cellular automata and evolutionary algorithms, which have been successfully used to model the processes governing the complex behavior occurring in nature.

In order to facilitate the development of the proposed design method, Emergent Engineering Design was implemented in a computer system called Emergent Designer. It is an integrated research and design support tool which applies models of complex systems to represent engineering systems and analyze design processes. Emergent Designer was used to conduct the empirical validation of the proposed design method for two classes of conceptual design problems in structural engineering. The extensive design experiments reported in this dissertation have shown that Emergent Engineering Design not only generates novel design concepts exhibiting remarkable structural shaping patterns but it also efficiently optimizes them.

1. INTRODUCTION

"One of the foundations for change in our society comes from designing. Its genesis is the notion that the world around us either is unsuited to our needs or can be improved. The need for designing is driven by a society's view that it can improve or add value to human existence well beyond simple subsistence. As a consequence of designing the world which we inhabit is increasingly a designed rather than naturally occurring one."

(John Gero)

In this dissertation, I introduce a new engineering design method called Emergent Engineering Design (EED). The method uses models based on complex systems to represent major elements of engineering design processes. Combinations of several types of complex systems have been investigated in modeling design representations, actual design processes, as well as their evaluation.

The goal of the research described in the dissertation was to establish a new design method that would satisfy two major engineering design objectives:

- Develop novel designs, and
- Optimize engineering designs.

EED consists of state-of-the-art models and procedures that are inspired by complex phenomena occurring in nature. Research that I present in this dissertation is an attempt to formulate a design method that is conceptually coherent and inspired by the fundamental processes occurring in nature. Nature has arguably created the most fascinating designs known to humankind (French 1994). On the other hand, the fundamental processes governing the complex behavior occurring in nature have been successfully modeled using various complex systems. It is my belief that this inspiration can be effectively used in solving a broad range of engineering design problems.

1.1. Motivation

The motivation for this work comes from my recent interests in the area of evolutionary design, dynamical systems, and cellular automata. I frequently observed, in various design experiments involving evolution-guided generation of design concepts of steel structural systems in tall buildings, formation of emergent and novel structural shaping patterns (Kicinger et al. 2002). Countless examples of emergent phenomena generated by complex systems have been also reported by researchers from various disciplines (Gero 1992; Ilachinski 2001). However, little has been done in terms of building a coherent engineering design method based on complex systems that emphasizes both aspects of a design process: novelty and optimization.

Engineering design research has been focused on the development of various design theories, methodologies, methods, tools, and procedures for a long time (Newsome et al. 1988). The design methods have been subsequently used by engineers to more efficiently design artifacts. However, as the artifacts have grown in complexity, the need for new methods has become obvious. Also, in a nowadays world, increased competition and globalization require organizations to reexamine traditional product development strategies. Traditional methods focused exclusively on the quantitative/numerical optimality of produced artifacts, or their manufacturing processes, are no longer adequate. Creativity and innovation of designed artifacts provide organizations not only with a competitive advantage but are, in fact, a matter of their survival.

With the emergence of Information Technology, new design methods are being developed which are based on various computational models of design processes. However, up until very recently, computers in design were used mostly and merely for various analytical design activities conducted in the final part of the engineering design process, namely in the detailed design stage (Arciszewski and De Jong 2001). Today, we are finally witnessing the emergence of new design support tools applicable both in the conceptual and detailed design stages, i.e. tools that are suitable for both generation of novel design concepts and their subsequent optimization. In order to fully benefit from this progress, these new tools require new design methods and computer tools.

1.2. Research Justification

Our understanding of engineering design has been recently undergoing significant changes. Not only was the previous focus on acquiring engineering knowledge replaced by the processing and utilization of available knowledge using advanced computer systems and design software, but also the traditional simplified (usually linear) models of physical and mathematical interactions are now being substituted with nonlinear models, which more accurately represent real-world phenomena (Arciszewski et al. 2003; Thompson 1999). Currently, we are witnessing another emergent trend of replacing complicated models with distributed, or parallel, models based on simple rules/programs and interactions among elements that can also generate very complex behavior (Wolfram 2002). Thus, even though the models studied in the conceptual design phase are becoming more and more complex, it is possible that this complexity can be modeled using only very simple rules and programs. Hence, the complex systems approach to conceptual design seems to be a plausible way of capturing the complex nature of the design process and may enable us to use simple mathematical and computational models to simulate this process. Also, presently available computing power opens new possibilities of designing and modeling complex engineering systems and their dynamic evolution.

Complex systems are dynamical systems that consist of large numbers of mutually and, typically nonlinearly, interacting parts. One of the characteristic properties of complex systems is their emergent behavior. Complex systems can also be characterized by their adaptive behavior, i.e. an underlying mechanism to adapt and survive in uncertain environments. From an engineering point of view, it is important to ensure that engineering designs can adapt to changing design requirements and constraints because that guarantees their robustness, a required property of almost all engineering products (Gen and Cheng 2000).

Besides adaptation, another important and inherent property of complex systems is their spatio-temporal evolution. The process of evolution can be understood in a very broad sense as a gradual transformation of a system over time, but it also has its narrower meaning in biology, namely as a Darwinian evolutionary system (Darwin 1859). The Darwinian concept of evolution by means of natural selection provided inspiration for researchers in evolutionary computation (EC) and resulted in a family of modern heuristic search algorithms called evolutionary algorithms (EA). The Darwinian evolutionary system is also one of the prominent examples of a complex adaptive system.

Computationally simulated evolution is an important basis for understanding life (Holland 1975), but it has also been applied for studying and solving problems in other disciplines. Among newly developed computational paradigms, evolutionary computation is now recognized as particularly appropriate for various traditional and novel computational applications in engineering. This paradigm has already been applied to many engineering design problems including both optimization as well as creative design problems. This application has been done with much success, and even a subfield within EC community, called evolutionary design, has emerged (Bentley 1999a; Parmee 1999).

Evolutionary design support tools allow researchers and engineers to produce thousands or even hundreds of thousands of feasible design concepts in a relatively short period of time. On the contrary, human designers tend to limit the range of design concepts being considered to only a few alternatives. The evolutionary design process is not merely a random search process; it is a fitness guided generation of design concepts. Thus, two goals that are extremely important in engineering design are achievable. First, one can build a collection of design points in a given representation space, or a so-called 'big picture' of a design representation space, and thus acquire a significant amount of design knowledge. Second, an evolutionary guided intelligent search within this design space can support discovery of novel designs (Arciszewski et al. 2003).

Another important type of an evolutionary complex adaptive system is a coevolutionary system. Here again, the inspiration comes from biological processes encountered in many natural ecosystems. Coevolutionary processes can be modeled by a class of coevolutionary algorithms. Initial ideas of using coevolution as an optimization procedure were formulated by Axelrod (1984; 1987) in the context of competitive fitness functions. Potter and De Jong (1994) proposed a cooperative coevolutionary model and developed a cooperative coevolutionary evolutionary algorithm (CCEA). Complex adaptive coevolutionary approaches (both competitive and cooperative) have strong potential in engineering design but almost no work has been done in this area.

Cellular automata (CAs) are examples of complex systems with enormous and still unexplored potential to develop novel designs. They are one of the simplest mathematical and computational representations of complex systems. As such, they can be used as useful idealizations of the dynamical behavior of various systems. They appear to capture many essential features of complex self-organizing cooperative behavior observed in real world systems. CAs have been devised to model complex systems and processes consisting of a large number of identical, simple, locally interacting components. CAs can be used to study pattern formation and gain some insight into self-organization processes (Ilachinski 2001). The CAs research has generated great interest over the last forty years because of their ability to exhibit complex patterns of behavior using a set of simple underlying rules. However, very little has been done in terms of their application to engineering design problems.

The significance of the CAs in engineering design can be explained considering several facts. First, they can inherently model spatial relations of various elements in an engineering system. Second, they can explicitly represent local interactions among elements of an engineering system. Third, CAs are known to produce various kinds of emergent behavior. This property is highly relevant in many engineering design problems, e.g. novel structural shaping patterns. Finally, it is a fact that even designers of complex and sophisticated engineering systems (bridges, tall buildings, etc.) use only a small set of design/decision rules to develop design concepts. This set of design/decision rules can be represented by the transformation rules of a CA. From the engineering perspective, CAs can be viewed as "black-box" concept generators that, given some input and a representation of an engineering system, use simple transformation rules and local interactions among design elements to produce an output (a final design concept), that possibly contains some interesting patterns.

The behavior of complex systems is, as their name well suggests, very difficult to describe formally in terms of traditional mathematical models. On the other hand, as it was shown in (Wolfram 2002), the apparent complexity of behavior does not imply the complexity of underlying mechanisms causing this complicated behavior. On the contrary, in many cases the underlying rules can be extremely simple. Some theoretical approaches have also been proposed to describe this seemingly random behavior. They include dynamical systems theory and chaos theory (Alligood et al. 1996) which provide quantitative models of studying complex nonlinear phenomena well founded in traditional mathematics. They identify and formally describe classes of dynamical behavior (fixed point behavior, periodic orbits, chaotic orbits), and provide appropriate measures to quantify this behavior, e.g. Lyapunov exponents. On the other hand, Wolfram (2002) provides qualitative and computational models based on the iteration of simple programs.

Using dynamical systems theory and chaos theory to describe evolutionary processes has profound justification. First, evolution is an inherently dynamical process during which individuals change in both space and time. We may express the complex behavior of an evolutionary system using available mathematical models, e.g. Lotka-Volterra (Lotka 1925; Volterra 1926) model of predator-prey evolution. Also, evolutionary algorithms can be formally described using dynamical systems theory (Vose 1999b). Second, as it has been reported by some researchers (Packard 1988), the emergent patterns in complex systems occur at the stages of evolution when systems undergo phase transitions, which can be well described using bifurcation theory, a subfield of dynamical systems theory. It has also been argued that the most interesting patterns occur when the system is about to change its behavior to chaotic, that is at "the edge of chaos" (Packard 1988). It has already been discovered by design researchers that novel design concepts emerge when a significant change occurs; For example, a change in the representation space during the constructive induction process (Arciszewski et al. 1995).

This last interpretation is consistent with a much broader philosophical discussion concerning the process of discovery. Charles Peirce (1998), one of the most prolific American philosophers, has proposed the third kind of inferential process, called abduction, which he claimed was involved in the processes of discovery. In some modern interpretations of Peirce's abduction, the process of mental activity involved in discovery is considered to be chaotic in nature and consisting of various free associations that on the surface seem disorganized and unsystematic (Koestler 1990). Koestler claims that flashes of insight, as suggested by Peirce to accompany episodes of discovery, can be explained by what he terms bisociation. Bisociations represent intersections of two different frames of reference, or knowledge representations, or knowledge from two domains, which can be modeled as bifurcations. Yet another outlook on the process of discovery is presented by Singer (1995). He regards the process of discovery as an emergent phenomenon and claims that the new insights somehow emerge as a result of the nonlinear aggregations of an imaginably complex collection of interacting neural elements in the brain. Similar opinion is presented by Crutchfield (1994), who describes discovery as a result of synthesis of tools from dynamical systems, computation, and inductive inference. The synthetic interpretation of the process of discovery is very close to the tradition of synesthesia (from Greek, *syn* = together + *aisthesis* = perception), proposed by Leonardo da Vinci, one of the greatest discoverers and creators in the history.

As it has been argued above, complex systems method for engineering design can be justified from various perspectives and might bring broader fundamental understanding of design processes as well as a new generation of design support tools. Using the new method, it should be possible to model and implement the processes of creative design in many areas of engineering that traditionally use strictly defined sets of codes, or sophisticated rules due to complexity of their domains and developed through incremental experience. This method considers both aspects of engineering design, i.e. novelty and optimality. Potential for novelty in design is introduced by using state-of-the-art representations of engineering systems and mechanisms to generate design concepts. Optimality can be achieved by using evolutionary and coevolutionary search processes guided by the fitness of design concepts. This method might also be useful for building a global (holistic) picture of a given engineering domain and hence provide significant amounts of new domain knowledge, which can be subsequently utilized. Available theoretical foundations should provide useful mathematical models and quantitative methods of analysis of engineering design processes.

1.3. Organization

The remainder of the dissertation is organized as follows:

Chapter 2 contains background material that is relevant for understanding the rest of the dissertation. It contains state-of-the-art overviews of the disciplines related to this research. It also serves as a helpful reference material to which I frequently point to in the remainder of this dissertation.

Chapter 3 introduces Emergent Engineering Design, the major objective of this dissertation, and provides a description of its assumptions. It also discusses the structure of the argument presented in this dissertation in the form of research questions and hypotheses and offers a detailed description of the validation methodology.

Chapter 4 proposes novel design representations based on models of complex systems which are investigated in this dissertation. Several types of design representations based on cellular automata are introduced and described in detail.

In chapter 5, Emergent Designer, a unique design support tool, is introduced and presented. It is an integrated research and design support tool which implements Emergent Engineering Design, the design method proposed in this dissertation.

Chapter 6 begins the experimental part of this dissertation. It investigates specific instances of complex systems, namely cellular automata, as design concept generators of structural systems and subsystems. This chapter focuses on the aspects of novelty in design processes and does not discuss evolutionary based optimization mechanisms.

Chapter 7, on the contrary, focuses exclusively on design optimization issues. It describes evolutionary based optimization using standard parameterized representations. Various design experiments reported in this chapter focus on optimization of several structural systems and subsystems. The experimental results are accompanied with the quantitative analysis and presentation of the best design concepts.

Chapter 8 presents a combined approach, called morphogenic evolutionary design, in which generative representations based on cellular automata are evolved using evolutionary algorithms. It describes results of applying this new design method to the same problems considered in the previous two chapters and compares the outcomes. Also, the qualitative and quantitative analysis of the results is presented.

Extensive experimental work described in chapters 6 - 8 is divided into sections. All experimental parameters and their values as well as obtained results are reported in individual sections while the summary of major findings can be found at the end of each section.

In the final chapter 9, conclusions and discussion on the findings are presented. Also, the contributions of this dissertation to the field of engineering design are discussed. Finally, some recommendations for the most promising paths of future research are offered.

Appendix A provides a chronological classification of applications of evolutionary computation in structural engineering. Relevant publications since the beginning of the field were classified with respect to the application domain and the structural problem addressed.

Appendix B contains a complete collection of 256 design concept generated using elementary cellular automata with periodic boundary conditions while Appendix C shows another set of design concepts of wind bracing systems developed using elementary cellular automata but this time with nonperiodic boundary conditions.

A suggested sequence for reading this dissertation consists of the following chapters:

Chapter 1 - Introduction

Chapter 3 - Emergent Engineering Design

Chapter 4 - Design Representations

Chapter 6 - Design Concept Generation Using Cellular Automata

Chapter 7 - Evolutionary Optimization

Chapter 8 - Morphogenic Evolutionary Design

Chapter 9 - Closure

Chapter 2 may be skipped at first reading. It provides background material to which I refer when necessary in the remainder of this dissertation. Chapter 5 is very technical and discusses the implementation details and information flow in Emergent Designer. It is recommended for readers interested in building modern design support tools.

2. BACKGROUND

"If I have seen further it is by standing on the shoulders of Giants"
(Isaac Newton)

This chapter contains a background material that provides some context necessary for understanding the rest of the dissertation. It describes recent developments in the disciplines relevant to Emergent Engineering Design, the design method proposed in this dissertation. The interdisciplinary character of this dissertation influenced the extensiveness of the review. It is aimed to provide an introductory material on the topics discussed in this dissertation to readers with different backgrounds and to present high-level overview of the current research developments in the relevant disciplines. It will also serve as a useful reference to which I will frequently point to in the remainder of this dissertation.

Figure 1 shows an organization chart of the background review included in this chapter. It is divided into five major parts which are presented in five subsections. First, a comprehensive survey of evolutionary computation (EC) and evolutionary design is provided in section 2.1. EC is one of the key components of Emergent Engineering Design. The goal of this survey is to show current research developments in this field with an emphasis on design optimization and creative design, two important objectives of engineering design addressed in this dissertation. The last part of section 2.1 provides a chronological overview of the applications of evolutionary computation in structural design and a discussion on the advantages and limitations of this approach when compared to traditional optimization methods. Finally, open issues in the field are discussed as well as the most promising directions of future research.

Section 2.2 contains a brief overview of another important component of Emergent Engineering Design, namely cellular automata (CAs). CAs are proposed in this dissertation as design concept generators which produce novel design concepts. The section introduces one-dimensional and two-dimensional CAs and discusses the richness of behavioral patterns generated by these simple instances of complex systems. CAs play an important role in a recently proposed the New Kind of Science (NKS) (Wolfram 2002) which is also briefly described.

In this dissertation, I propose an engineering design method which uses models of various complex systems to represent major elements of engineering design processes. I also propose dynamical systems approach to analyze design processes. Hence, a high-level introduction to dynamical systems, chaos theory, and complex adaptive systems is offered in section 2.3. It introduces definitions of dynamical systems and complex adaptive systems and provides some historical background. Also, some applications of dynamical systems and chaos theory in structural engineering are presented at the end of the section.

In order to better place the proposed design method in the context of state-of-the-art (SOTA) in engineering design, section 2.4 describes recent developments in this field. The particular emphasis is put on design theories, methodologies, and methods. The section also contains a classification of existing approaches to modeling engineering design processes. The classification is subsequently used to define Emergent Engineering Design.

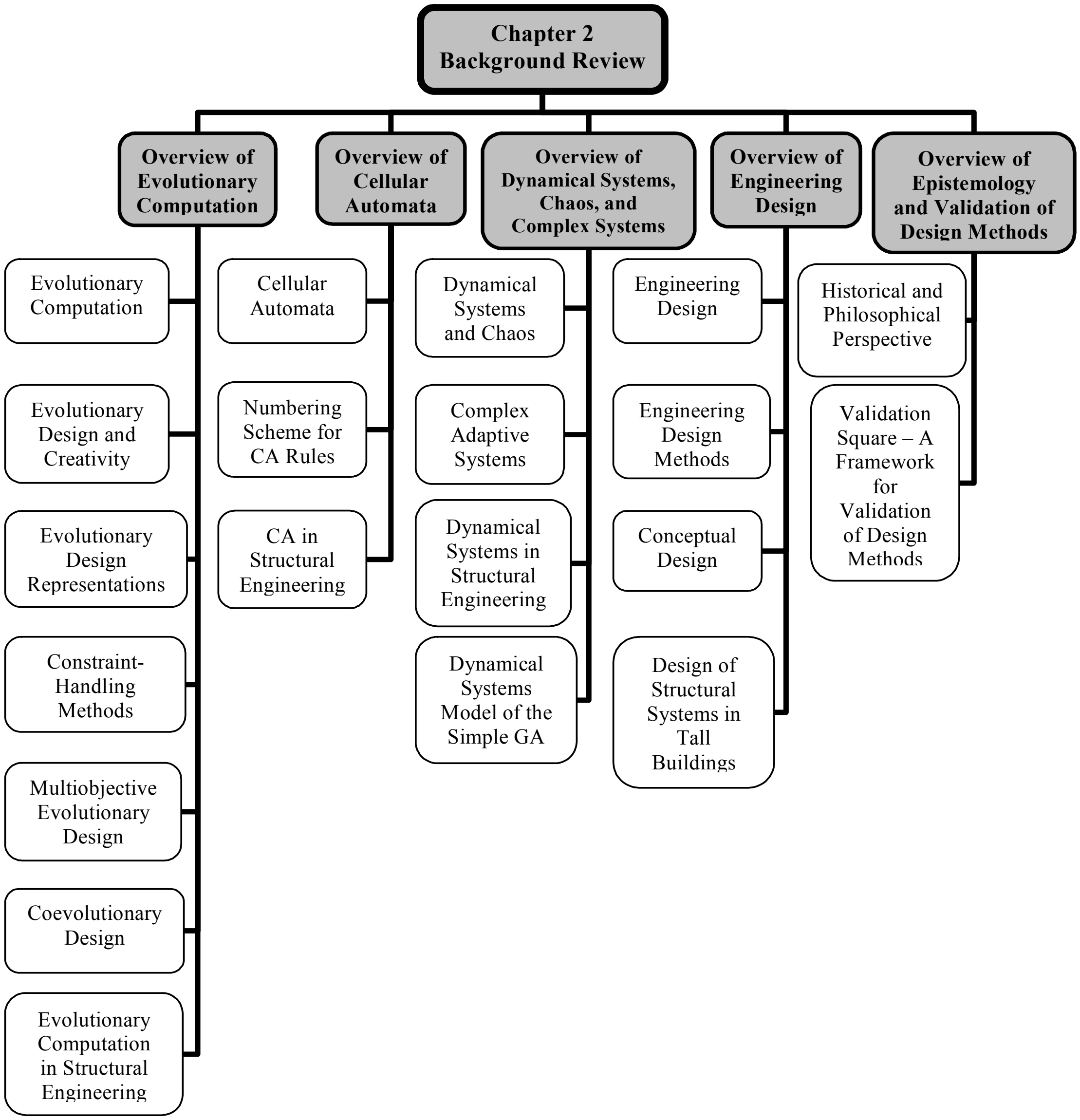


Figure 1. Organization of the background review

This dissertation claims to add a new scientific knowledge to the field of engineering design. But in order to make such claims, the new scientific knowledge must be first verified. Throughout the history of science, many different views have been presented on how to best validate scientific knowledge. They are briefly reviewed in the first part of section 2.5. The issue of validation of a new scientific knowledge is particularly relevant to the field of engineering design which is mainly concerned with open problems that involve both objective and subjective elements and have no single right answer. A recently introduced framework for validation of design methods, called Validation Square, is introduced in the second part of section 2.5. The framework has been used in this dissertation to validate Emergent Engineering Design.

The proposed design method has been validated empirically in the context of structural design problems. Thus, each section of this chapter contains a subsection that discusses relevance of the major ideas presented in the section to structural engineering and presents current research developments.

2.1. Overview of Evolutionary Computation

2.1.1. Evolutionary Computation

Evolutionary computation (EC) is a modern search technique which uses computational models of processes of evolution and selection. Concepts and mechanisms of Darwinian (1859) evolution and natural selection are encoded in evolutionary algorithms and used to solve problems in many fields of engineering and science.

Strong resemblance to biological processes as well as their initial applications for modeling complex adaptive systems (Holland 1975) influenced the terminology used by EC researchers. It borrows a lot from genetics, evolutionary theory and cellular biology. Thus, a candidate solution to a problem is called an *individual* while an entire set (or more accurately a superset) of current solutions is called a *population*. For some problem domains a population may be broken into several *subpopulations*. The actual representation (encoding) of an individual is called its *genome* or *chromosome*. Each genome consists of a sequence of *genes*, i.e. attributes that describe an individual. A value of a gene is called an *allele*. When individual solutions are modified to produce new candidate solutions they are said to be *breeding* and the new candidate solution is called an *offspring* or a *child*. During the evaluation of a candidate solution, it receives a grade called *fitness*, which indicates the quality of the solution in the context of a given problem. When the current population is replaced by offspring, the new population is called a new *generation*. Finally, the entire process of searching for an optimal solution is called *evolution* (Luke 2000).

Evolutionary Algorithms

Evolutionary algorithms are a family of population-based search algorithms that simulate the evolution of individual structures by interrelated processes of selection, reproduction, and variation. There is a variety of EAs that have been proposed and studied. They all share a common set of underlying assumptions but differ in the breeding strategy to be used and representation on which EAs operate.

Historically, three major EAs have been developed: evolution strategies (ES) (Rechenberg 1965; Schwefel 1965), evolutionary programming (EP) (Fogel et al. 1966), and genetic algorithms (GAs) (Holland 1975). These algorithms have been mostly used to evolve solutions to parameterized problem domains. On the other hand, the fourth major EA developed more recently, genetic programming (GP) (Koza 1992), has been used to evolve actual computer programs to solve a number of computational tasks (Luke 2000). There are also many hybrid models incorporating various features of the above paradigms, including the CHC algorithm (Eshelman 1991), the structured GA (Dasgupta and MacGregor 1991), the breeder GA (Mühlenbein and Schlierkamp-Voosen 1993), the messy GA (Goldberg et al. 1989), and many others.

From the engineering point of view, EC can be understood as a search and optimization process in which a population of solutions undergoes a process of gradual changes. This process depends on the fitness (a formal measure of perceived performance) of the individual solutions as defined by the environment (objective function).

Figure 2 shows the structure of a canonical EA. Before an actual evolutionary process begins, an initial population of individuals (solutions) is created. Traditionally, the initial population is created randomly but several other initialization techniques have also been used (e.g. starting from a set of previously known or arbitrarily assumed solutions). Next, each individual in the initial population is evaluated and assigned a fitness value.

Using the fitness scores, the selection mechanism chooses a subset of the current population as parents to create new individuals. When the selection mechanism uses bias toward individuals with better fitness, the created offspring will, more likely, have higher fitness. Once the set of parents has been selected, the new individuals are created by copying them and applying variation operators.

There are several commonly used selection strategies within EC community. *Fitness-proportional* selection (Holland 1975) normalizes the fitness values of all individuals in the population and assigns these normalized values as probabilities that their respective individuals will be selected. *Ranked selection* works by first ranking all individuals in the population by their fitness, and use these ranks, rather than actual fitness values, to determine selection probabilities of the individuals. A common form of ranked selection is a *linear ranking* (Grefenstette and Baker 1989; Whitley 1989) where individuals are first sorted in an increasing order according to their fitness values. Each individual is then selected with a probability based on some linear function of its sorted rank. Another popular selection strategy is a *tournament selection*. In this strategy, a pool of *n* individuals is picked at random from the population. Each of the individuals in the pool is selected independently and it might be the case that the same individual will be selected multiple times. Next, an individual from the pool with highest fitness value is selected to form the new population. This procedure is repeated as many times as necessary to create either an entirely new population or a subset of it. The pool size is a parameter that controls the magnitude of the selection pressure. Finally, the *truncation selection* chooses only a certain proportion of the best individuals in the population. This strategy is most popular within the ES community, where it is used in two basic flavors: (μ, λ) and $(\mu+\lambda)$ (Schwefel 1977). In the former case, the selection operates on the offspring population only, whereas in the latter case it selects individuals from a joint population of both parents and offspring.

The two most popular variation operators are *mutation* and *recombination*. Mutation acts on a single individual and works by applying some variation to one or more genes in the individual's chromosome (similar to a variation operator used in other search mechanisms like hill climbing or simulated annealing). Recombination, on the other hand, operates on multiple individuals (usually two) and combines parts of these individuals to create new ones.

The newly created individuals are evaluated and assigned fitness values. Then, either all or only a subset of the current population is replaced by these new individuals. If the entire population is replaced by the new individuals then the algorithm is called a *generational* EA. On the other hand, if only a subset of the original population is replaced then it is called a *steady-state* EA. Steps 3-6 of the canonical EA defined earlier are performed until an assumed stopping criterion is met, which is usually defined as an arbitrary number of generations or fitness function evaluations.

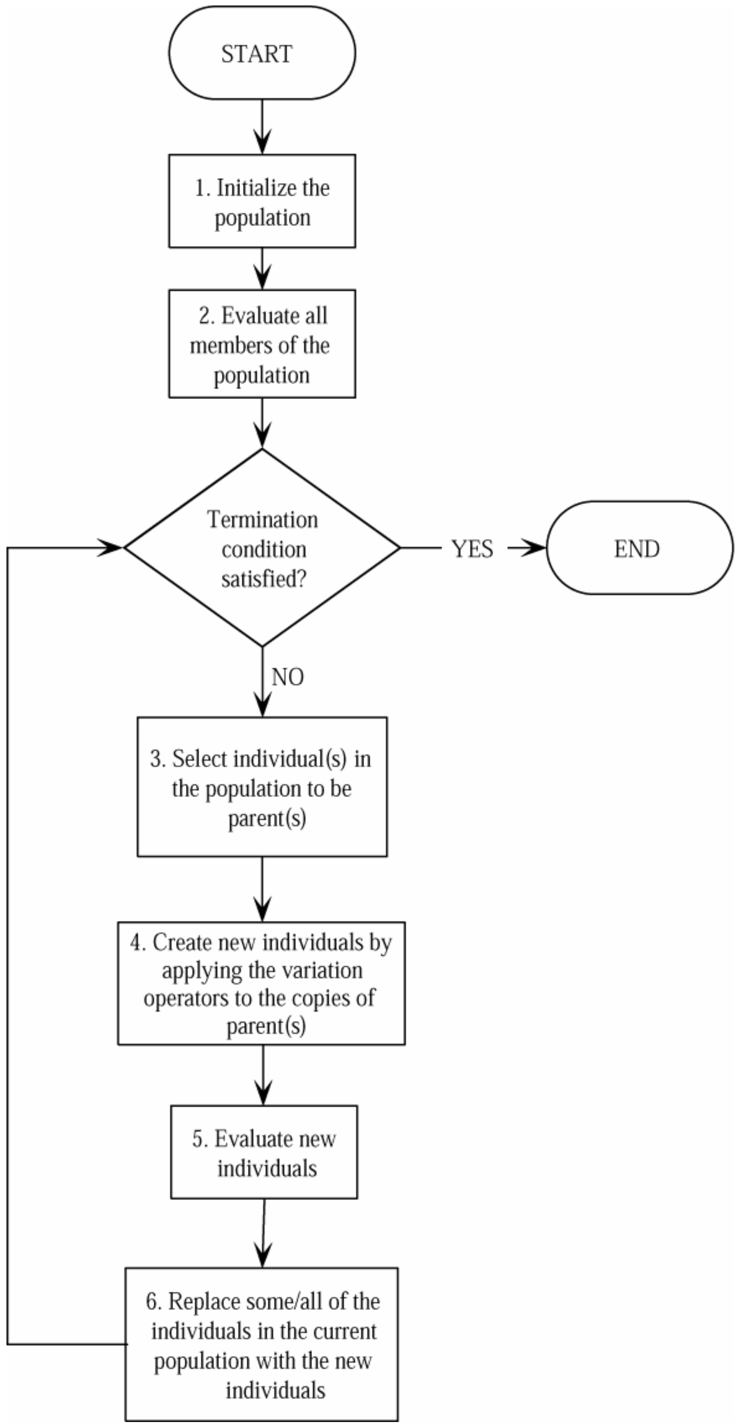


Figure 2. Structure of a canonical evolutionary algorithm

Evolutionary Computation and Engineering Design

This basic evolutionary process described above is called a 'simple evolutionary algorithm' in a sense that it contains the minimal set of features necessary to be a Darwinian evolutionary system. These simple EAs have surprisingly useful properties, primarily related to solving difficult global optimization problems. They perform well when applied to problems with nonlinear, stochastic, temporal, or chaotic components, where traditional optimization techniques, like gradient descent, hill climbing, and purely random search, are generally unsatisfactory. It is in this context that much of the work on engineering applications has taken place historically: using simple EAs for design optimization.

The three main issues in applying EAs to an engineering design problem are:

1. Selecting an appropriate representation for engineering designs.
2. Defining efficient genetic operators.
3. Providing an adequate evaluation function for estimating the 'fitness' of generated solutions (points in the search space).

An appropriate representation of an engineering system is one of the most crucial elements of evolutionary design. This issue is particularly important when creativity/novelty of designs produced by evolutionary processes is one of the major goals. The process of creating an efficient and adequate representation of an engineering system for evolutionary design is complicated and involves elements of both science and art. One has to take into account not only important aspects of understanding traditional modeling of an engineering system, but also relevant computational issues that include search efficiency, scalability, and mapping between a search space (genotypic space) and a space of actual designs (phenotypic space). A more detailed discussion of EA representations is presented in section 2.1.3.

An appropriate choice and implementation of genetic operators, i.e. mutation and recombination operators, and careful tuning of their rates is an important issue as it can have a big impact on the success of EAs. This issue has therefore been a subject of both theoretical (Spears 2000) as well as experimental investigations (Fairley 1991; Fogarty 1989; Schaffer and Eshelman 1991). Any particular implementation of a mutation or recombination operator is representation dependent. Thus, for example GAs with binary string representations uses the *bit-flip* mutation and 1-, or 2-point crossover, while ES with real-valued vectors use the *Gaussian* mutation and a recombination operator that swaps/averages parents' alleles. Genetic operators are primary sources of exploration in EAs. On the other hand, selection mechanisms provide EAs with exploitative power. Thus, by properly defining and controlling the variation mechanisms (genetic operators), one can achieve a higher level goal of finding "an effective balance between further exploration of unexplored regions of the search space and exploiting the regions already explored." (De Jong to appear).

Another important issue in successful application of EAs is to choose an adequate fitness evaluation function for a problem domain. Evaluation functions provide EAs with feedback about the fitness of each individual in the population. EAs use this feedback to bias the search process in order to improve the population's average fitness. Naturally, the details of a particular fitness function are problem specific.

Table 1 provides a description of all commonly used EAs in terms of decisions that are made during an implementation of a particular EA. It is a modified table initially proposed in (Arciszewski and De Jong 2001). The particular decisions are summarized in terms of attributes and their values. Using this characterization, it is then straightforward to describe a given EA, e.g. a GA or ES, and its relationship to other EAs (Arciszewski and De Jong 2001).

Table 1. Attributes describing commonly used EA implementations

Attribute			***Attribute Value(s)***				
1. Solution representation	*Encoding*		Binary	Real-valued	Graph-based	Computer code	Other
	Length		Fixed	Variable			
2. Population initialization	*Mechanism*		Random generation	Selection from a group of known solutions	User defined		
	Population size		1	Fixed	Variable		
3. Parent selection mechanism			Truncation	Ranking	Fitness proportional	Tournament	Uniform
4. Variation mechanism	*Mutation*	*Type*	Bit-flip	Gaussian	Subtree	User defined	
		Rate	0	Fixed	Adaptive	Random	
	Crossover	*Type*	N-point	Swap	Uniform	Subtree	User defined
		Rate	0	Fixed	Adaptive	Random	
5. Survival selection mechanism			Truncation	Ranking	Fitness proportional	Tournament	Uniform

Advanced Evolutionary Algorithms

Various modern trends in EC relax some of the assumptions found in the canonical EA. For example, in *multiobjective EAs*, a requirement of a single fitness value determining the quality of an individual is replaced by several independent fitness criteria. Another assumption of using a single evolving population is relaxed in *parallel*, or *distributed*, *EAs* as well as in c*oevolutionary algorithms* (CEAs). In a fairly popular model of a parallel EA, called the *island-model* EA (Cohoon et al. 1987), evolution occurs in multiple parallel subpopulations evolving independently with occasional 'migrations' of some individuals among subpopulations. CEAs typically use multiple subpopulations but additionally modify another fundamental assumption, namely that individuals are no longer evaluated independently of one another. Two common models of CEAs include *cooperative CEAs* (Potter and De Jong 2000), where the fitness of an individual is assessed through 'cooperation' with individuals from other subpopulations, and *competitive CEAs* (Angeline and Pollack 1993), where the fitness of an individual is determined by its competition against individuals from other populations. Coevolutionary EAs are discussed in more detail in section 2.1.6.

Next section presents a subfield of EC, called evolutionary design, which is directly related to engineering design problems. It also discusses the issues of creativity and emergence in engineering design processes.

2.1.2. Evolutionary Design and Creativity

Evolutionary design is a branch of EC that integrates ideas from computer science (evolutionary algorithms), engineering (design science) and evolutionary biology (natural selection) to solve engineering design problems (Bentley 1999c). Four major categories of problems considered by evolutionary design include evolutionary design optimization, creative evolutionary design, evolutionary art, and evolutionary artificial life forms.

Common attributes shared by evolutionary techniques, which are relevant to engineering design processes include (Parmee 1999):

- little, if any, a priori knowledge of the search environment
- excellent search capabilities due to efficient sampling of the design search space
- ability to avoid local optima
- ability to handle high dimensionality
- robustness across a wide range of problem classes
- provision of multiple good solutions
- ability to locate the region of the global optimum solution

Research on evolutionary computation in engineering design has a relatively long history. It was initiated in Europe in the early seventies by Rechenberg (1973) in the areas of fluid mechanics, pipe design and structural engineering. Early applications of EC in structural engineering (Hoeffler et al. 1973; Lawo and Thierauf 1982) used ES which evolved from structural optimization approaches in the early 1960's. Further significant progress in this area has taken place mainly during the last fifteen years. In the United States, Goldberg (1987; 1989) did the first application of GAs, which emerged from the machine learning community, in engineering optimization in the area of complex gas pipeline systems. Just about the same time, in the late 80's and early 90's, many researchers started applying this new optimization method to a large spectrum of engineering design problems. Current state-of-the-art (SOTA) reviews are provided in (Arciszewski and De Jong 2001; Bentley 1999a; Bentley and Corne 2002; Chawdhry et al. 1998; Coello Coello et al. 2002; Cvetkovic and Parmee 1999; Dasgupta and Michalewicz 1997; Gen and Cheng 1997; Gen and Cheng 2000; Parmee 1999; Parmee 2001; Parmee 2002).

Creative Design

Evolutionary design optimization and creative evolutionary design are the two categories of evolutionary design that are particularly relevant to civil and structural engineering applications. From a computational point of view, the dividing line between the two categories is not sharp and is mostly related to the potential of achieving novelty/creativity during the processes of generating design concepts as well as properties that novel/creative designs need to possess. For Gero (1996) creativity in design "is not simply concerned with the introduction of something new into a design, although that appears to be a necessary condition for any process that claims to be labeled creative. Rather, the introduction of 'something new' should lead to a result that is unexpected (as well as being valuable)." Gero concludes that an evolutionary design process is creative when it explores not only values of attributes (decision variables) within individual design spaces but also evolves the number of these attributes, i.e. when changes in the representation space occur. Similarly, Boden (1992) suggests that achieving creativity is only possible by going beyond the bounds of a representation, and by finding a design that could not have been defined by that representation. The same concept was explored by Arciszewski and co-workers in the context of Inferential Design Theory (Arciszewski and Michalski 1984) and constructive induction (Arciszewski et al. 1995). A detailed discussion of commonly used representations in evolutionary design, including generative representations supporting creative design processes can be found in section 2.1.3.

Less restrictive definition of creativity in design was given by Rosenman (1997). He suggested that the distinguishing feature of all creative evolutionary design systems is the ability to generate entirely new designs starting from little or almost no knowledge (for example when starting with random initial conditions), and being guided throughout the evolutionary process only by performance criteria.

Evolutionary Design and Theory of Inventive Problem Solving

Creativity in evolutionary design can also be analyzed from a broader perspective, namely based on the theory of inventive problem solving (TRIZ) introduced by Altshuller (1969; 1999). Altshuller discovered that the evolution of engineering systems is not a random process, but is governed by a class of paradigms. These paradigms can be subsequently used to develop a system considering its technical evolution, i.e. by determining and implementing innovations. Altshuller introduced five levels of innovation in the context of an engineering design problem (Arciszewski et al. 1995):

1. **Selection**

   "A design concept is selected from a group/class of known concepts in a given engineering domain."

   This level of innovation corresponds to **an EA using only selection operation** and that is initialized with a population of known design solutions, rather than randomly generated ones.

2. **Modification**

   "A design concept is produced as a combination and/or modification of known design concepts from a given domain. The modification process can be performed either deterministically or using a random generation process."

   This paradigm is equivalent to **an EA searching for an optimal solution in a parameterized representation space** of a class of engineering designs. Rosenman's definition of creativity in design is most closely related to this paradigm and hence it becomes obvious that his prerequisites of creativity are fairly weak when compared to Altshuller's innovation taxonomy.

3. **Innovation**

   "A design concept is produced as a combination of known concepts from a given domain and other domains."

   This paradigm can be best represented as the **island-model EA** where various populations of designs evolve independently and occasionally exchange some individuals through a migration process. The migrations can model injection of knowledge from other domains to a particular engineering domain.

4. **Invention**

   "A design concept is produced as a combination of known concepts from a given domain and new concepts based on a new technology, which have been recently introduced."

   EA can achieve this level when it evolves not only the values of attributes but also the attributes themselves (Rosenman and Gero 1999). In other words, it can use various transformation operators (Arciszewski et al. 1995) for a representation space including attribute addition (introduce new attributes/genes to the representation space), attribute elimination (removing unimportant attributes), attribute abstraction (combining attributes into larger units, or components, and subsequently exploring the component based representation (Bentley 2000)), and attribute construction (creating new attributes by a simple or complex transformation of the initial attributes). This level of innovation is most closely related to Gero's definition of creativity in design as well as changes in the representation space introduced in the constructive induction process (Arciszewski et al. 1995).

5. **Discovery**

"A design concept is produced as a combination of known concepts from a given domain and new concepts based on new scientific principles."

This highest level of innovation in Altshuller's taxonomy can most likely be achieved by evolutionary design processes. However, special types of representations, namely the generative representations (Hornby 2003) (described in section 2.1.3), seem to be necessary to accomplish it. Generative representations use compact representations (genotypes) of existing design knowledge and mappings that translate these representations to actual designs (phenotypes). The mappings can reuse elements of the representations during the process of translation. Thus, the compact representations can be thought of as storing existing knowledge on a given engineering domain, whereas mappings correspond to new scientific principles that can transform the known concepts to new, and possibly creative, design concepts. The mappings are usually simple programs that take the compact representations as input and produce actual design concepts as output. Despite their simplicity, they can generate designs that can be defined as creative (Bentley and Kumar 1999). Recently, Wolfram (2002) suggested that all scientific principles and natural processes can be modeled in terms of simple programs that can nevertheless produce complex behavior. **EA using the generative representations** will search both the space of compact representations and the space of simple transformation programs (scientific principles) and will generate creative design concepts.

The first two paradigms, i.e. selection and modification, can only produce *routine designs*. In both cases, no changes occur in the representation space (Arciszewski et al. 1995). The last three paradigms, i.e. innovation, invention and discovery, can generate *novel/creative designs*. In all these cases, changes in the representation space do occur (Arciszewski et al. 1995).

Emergence

Emergence is an important property which is closely related to creativity in design. Gero (1992) defines emergence as "a process of making features explicit, that were previously only implicit." He also suggests that emergence plays an important role in introducing new attributes to the representation space (Gero 1996). Emergence can also be easily recognized through the visual examination of representations of structures, for example of structural patterns of steel structural systems in tall buildings (Kicinger et al. 2002).

The notion of an *emergent concept* generation has also been introduced by Arciszewski et al. (1995) as a part of a constructive induction process that was originally proposed in the field of machine learning. An *emergent design concept* is defined as a constructed attribute (representing an unknown design concept) whose introduction may simplify and improve effectiveness or quality of a design process. A constructed attribute is derived from the initial attributes by an application of constructive induction operators. It is usually more abstract than the attributes from which it was derived.

Integrated Design

Most applications of evolutionary methods in civil and structural engineering were focused on a detailed design stage of a design process, where the objective was to find the optimal configuration of attribute values for a previously selected and parameterized design concept. Thus, only routine design concepts could be generated, even though they were optimized with respect to some objective. An overview of the SOTA in evolutionary design applications in civil and structural engineering can be found in section 2.1.7.

There has also been some work in applying evolutionary design methods at the conceptual stage of an engineering design process, where the emphasis is on the generation of novel and original design concepts, and not on finding the globally best solution in terms of numerical values in the context of a specific design concept. Gero and Schnier (1995) worked on the evolution of a design knowledge representation, using genetic algorithms and Rosenman and Gero (1999) used genetic engineering to evolve architectural floor plans. Arciszewski et al. (1999) used evolutionary computation to produce creative designs. Bentley (1999b) developed a generic evolutionary design system, which was able to evolve a range of various designs from scratch. The system performed evolutionary design with an emphasis on the evolution of creative design concepts rather than their optimization.

The concept of integrated design utilizing various forms of evolutionary computation at each stage of a design process as well as incorporating designer's knowledge and intuition within the search and exploration process has been pioneered by Parmee (1995; 2001). In the mid-90's, this research was initiated on the utility of evolutionary/adaptive search within the generic domain of an engineering design process as a whole. Parmee, following Pahl and Beitz (1996), distinguishes three major stages of an engineering design process: *conceptual design, embodiment design,* and *detailed design*. He considers conceptual design as "a search across an ill-defined space of possible solutions using fuzzy objective functions and vague concepts of the structure of the final solution." Embodiment design operates with a selected (during the conceptual design stage) initial design configuration and aims to further specify the subsets forming the whole system. Design decisions at this stage are made based on both qualitative and quantitative criteria which usually are difficult to be formally defined using mathematical models and hence difficult to include in a scalar objective (fitness) function. Finally, at a detailed design stage, design decisions are made based on solely quantitative criteria which are well described by mathematical models, even though they may be computationally expensive and may require complex analysis techniques. Contrary to traditional and simplified definitions of engineering design process which assume little or no interaction between the stages (Pahl and Beitz 1996), Parmee argues that considerable overlaps exist among the three stages and they should be taken into account in the integrated design model. He suggests that a model of a design optimization process should be considered to "represent a long-term, highly complex process commencing with high-risk conceptual/whole-system design and continuing through the uncertainties of embodiment/preliminary design to the more deterministic, relatively low-risk stages of detailed design and the eventual realization of an optimal engineering solution."

The objective of Parmee's integrated design was to develop co-operative frameworks involving a number of evolutionary/adaptive computing techniques and integrate them with each stage of the engineering design process. During this research, various forms of evolutionary computation were considered in the context of integrated design, including structured genetic algorithms (Dasgupta and MacGregor 1991), GAANT algorithms (Parmee 1996), and ant colony algorithms (Bilchev and Parmee 1995; Colorni et al. 1992) as well as constraint satisfaction (Michalewicz et al. 1996). Next, Parmee investigated evolutionary computation in the context of searching "whole-system design hierarchy" described by both nominal and numerical attributes (1998a), and he applied it to designing hydropower systems (1998b). Later, Vekeria and Parmee (1996) proposed the use of evolutionary computation in conceptual design of structural systems, including the determination of the topology of their members. Recently, he has been focused on the "innovative conceptual design" in the context of variable mutation cluster-oriented genetic algorithms (vmCOGA) and successfully used them in the area of aerospace engineering (Bonham and Parmee 2000).

2.1.3. Evolutionary Design Representations

Representations in engineering design incorporate both representation of an *artifact* being designed as well as representation of a *design process*, i.e. a process by which the design is completed. The line distinguishing artifact representation and design process representation is often blurred. Building a representation of an artifact is similar to the process of its numerical/mathematical modeling in engineering science. It is, however, significantly broader because it encompasses much more knowledge than can be set into mathematical formulas and their numerical realizations. Generally, a representation of a designed artifact should describe its function, form, intent, legal requirements, etc. Advances in computer science, and evolutionary computation in particular, made it possible to use symbolic representations to describe objects, attributes, relationships, concepts, etc. Thus, it is now possible to capture more abstract and conceptual design knowledge (Dym 1994).

A *representation* of an engineering design is a computational description of an engineering system (that usually does not yet exist) expressed in terms of attributes (Arciszewski et al. 1995). In the most straightforward EC representation, each gene corresponds to an attribute and represents a dimension of the search space. Each such dimension can have an appropriate set of values (discrete or continuous) that a feature represented by this dimension can take on. In the simplest case, these representations use binary genes denoting the presence, or absence, of a feature. In such representations each individual consists of a fixed-length binary string of genes, or a genotype, representing some subset of a given set of features. Often, in complex engineering applications, multi-valued attributes are more natural to use (Arciszewski and De Jong 2001).

A *representation space* for an engineering design is a multidimensional space spanned over attributes that are used to describe an engineering design (Arciszewski et al. 1995). Attributes can be *symbolic* (when they take values from an unordered or partially ordered set) or *numerical* (when they take numerical values representing quantities or measurements). Symbolic attributes that take values from an unordered set are called *nominal* attributes; when they take values from a partially ordered set, they are called *structured*. Design concepts are typically described in terms of symbolic attributes. Numerical attributes are used for a detailed description of a design.

A *design concept* is understood as a description of a future engineering system, actual or abstract, in terms of a feasible combination of symbolic attributes and their values. After a conceptual design process is completed, a given design concept is used next in the detailed design process to produce a detailed design. A *detailed design* is understood here as a detailed description of a future engineering system in terms of both symbolic and numerical attributes (dimensions, weights, etc.) (Arciszewski et al. 1995).

Optimality vs. Creativity

A choice of a particular representation of an engineering system for an evolutionary design process is highly influenced by the designer's goal, i.e. whether the emphasis is on optimality in terms of numerical values in the context of a specific design concept, or on generation of creative design concepts. When the focus is on finding an optimal design, designers' attention is usually restricted to a particular concept or at most several concepts of existing designs. In this case, design representations usually take a form of *parameterizations* of an engineering system, or its parts. The parameters are then encoded as genes and their alleles are evolved using evolutionary algorithms in order to find the best design that maximizes (or minimizes) given objective(s). Thus, for strictly engineering optimization problems, representations should be *direct* (i.e. they should encode possible solutions) and *parameterized* (allowing only for slight variations). Traditional representations frequently used in engineering optimizations problems, like binary representations, integer representations, and real-valued representations can be included in this category. Additionally, representations used in optimization problems usually incorporate domain knowledge, to smaller or larger extent, in order to make the search more efficient.

Creative evolutionary design requires, however, more general and usually more complex representations. Representations that have been used in creative design are diverse but nevertheless share some similarities. Typically, phenotype representations are quite general and thus capable of representing large numbers of alternative shapes, forms, or morphologies (forms together with structures) (Bentley 1999c). They range from direct representations, as in voxel-based representations (Baron et al. 1997) or array-based representations (Kane and Schoenauer 1995; Kane and Schoenauer 1996), to highly *indirect* representations, i.e. representations that do not encode solutions but rather rules on how to build these solutions. The most popular examples of indirect representations are grammars (Roston 1994), trees (Bentley 1996; Funes and Pollack 1999), shape grammars (Grabska 1993; Schmidt and Cagan 1998; Shea et al. 1997; Stiny 1980), graphs and matroids (Shai 2001), cellular automata (Frazer 1995; Hajela and Kim 1999), L-systems (Coates 1997; Hornby 2003; Jacob 1994), and embryogenies (Bentley and Kumar 1999).

Selecting Appropriate Design Representations

Gen and Cheng (2000) discuss five major requirements for designing good representations (genotype-phenotype mappings) for evolutionary design problems:

1. **Non-redundancy**

   "The mapping between encodings and solutions should be 1-to-1."

   There should be a unique pairing of each element of a genotypic space with a corresponding element of a phenotypic space. Out of all three possible cases, the *1-to-n* mapping should be particularly avoided because it corresponds to multiple phenotypic representations of the same genome. In this case, an additional procedure would have to be employed to determine the actual phenotype.

2. **Legality**

   "Any permutation or combination of an encoding corresponds to a solution."

   It is important to distinguish between two basic concepts: *infeasibility* of a solution and its *illegality*. Infeasible solution means that a phenotype decoded from a genotype lies outside of a feasible region (defined by the constraints) in the phenotypic space. Illegal solution means that a genotype does not represent any phenotype for a given problem.

The implicit significance of the legality requirement is that it implies that standard genetic operators can be easily applied to a representation satisfying this requirement.

3. Completeness

"Any solution has a corresponding encoding."

This requirement guarantees that any phenotype has a corresponding genotype, and hence it is accessible to genetic search.

4. Lamarckian property

"The meaning of alleles for a gene is not context dependent."

This requirement "concerns the issue of whether or not one chromosome can pass on its merits [learned traits] to future populations through common genetic operators" (Cheng et al. 1996). If the **meaning** of alleles for a gene is interpreted in a context-dependent manner, as in the *non-Lamarckian* case, the offspring usually inherit nothing from parents. Generally, the representation should have the Lamarckian property so that offspring can inherit goodness from parents.

5. Causality (also known as Continuity)

"Small variations on the genotype space due to mutation imply small variations in the phenotype space."

This requirement focuses on the preservation of neighborhood structures. The appropriate choice of genotype-phenotype mapping in combination with the genetic operators is important for a successful evolutionary search process (Sendhoff et al. 1997). For a successful introduction of new information by an operator, the operator should preserve the neighborhood structure in the corresponding phenotype space. Search processes that preserve the neighborhood structure are said to exhibit *strong causality*.

Taxonomy of Representations

Representations used in evolutionary design have been classified with respect to many different criteria. Table 2 presents a compilation of classification schemes in which attributes and their values correspond to various categorizations of evolutionary design representations proposed by several researchers (De Jong to appear; Hornby 2003; Popovici 2003).

Table 2. Classification of EA representations

Attribute	*Attribute value(s)*	
EA level	Genotypic	Phenotypic
Structure	Linear	Nonlinear
Length	Fixed	Variable
Change during evolution	Static	Dynamic
Encoding scheme	Direct	Indirect
Accuracy of solution specification	Parameterization	Open-ended
Ability to reuse encoding	Non-generative	Generative
Genotype-phenotype correspondence	Explicit	Implicit

One of the most important representational issues is the choice between a *genotypic* and *phenotypic* representation. This issue has some important consequences not only for EC in general but also for evolutionary design. When one decides to use a genotypic representation (as it is the case in the canonical GA) then an appropriate genotype-phenotype mapping has to be constructed, hopefully satisfying all five major requirements stated earlier. A particular attention has to be paid to satisfy the causality requirement. The lack of correlation between variation at the genotype level and variation at the phenotype level can cause serious problems (De Jong to appear). When a genotypic representation is used, mutation and recombination operate at the genotypic level while the fitness evaluation and selection are performed at the phenotypic level. One of the advantages of using genotypic representations is the ability to reuse standard genetic operators for multiple problem domains.

Alternatively, one can just use phenotype level encodings (as it is the case in the canonical ES) to both explore and exploit a design space. The significant advantage of this approach is that no mapping between genotype and phenotype is necessary and hence all five requirements stated earlier are automatically satisfied. One can focus on achieving useful exploration only at the phenotypic level. The disadvantage of phenotypic representations is that the genetic operators become problem dependent and have to be carefully crafted for each individual problem domain (De Jong to appear). Phenotypic encodings have been widely used within the ES community and applied to many engineering optimization problems.

A *structure* of an evolutionary design encoding is another relevant criterion. Generally, representations can be divided into *linear* and *nonlinear*. A linear representation can be thought of as a 1-dimensional representation usually in a form of a string (binary, real-valued, integer-valued), list, etc. Nonlinear representations, on the other hand, have 2-, or higher- dimensional structure, e.g. trees, arrays, etc.

Another distinguishing property of evolutionary design representations is their *length*. They can be divided into two groups: *fixed-length* and *variable-length* representations. The length of a genome is constant during an entire evolutionary process when fixed-length encodings are used. It is not the case with variable-length representations where an individual can be represented by a genome that changes its length every generation. Consequently, a population may consist of individuals whose genomes have different lengths. Fixed-length representations have been widely used in evolutionary design optimization while variable-length representations have been applied to creative evolutionary design (Bentley 1999a).

Depending on whether, or not, a representation can change during an evolutionary design process, one can divide representations into *static* and *dynamic*. This is a more general classification than the one based on a change of the length of a genome because it considers not only a time-dependent change of the length of a genome but also time-dependent changes made to its structure.

Direct representations encode essentially the actual design concepts, while *indirect* representations encode rules on how to construct these concepts. Again, direct representations are used mostly for evolutionary design optimization and indirect encodings for evolving creative design concepts (Hornby 2003).

In the case, when the topology of a design is established in advance and specified in sufficient detail, i.e. it is parameterized; the representation is called a *parameterization*. On the other hand, when the topology of a design is changeable then the representation is called *open-ended*.

Representations that can reuse some parts of an encoded design from a genotype during the phenotype construction phase are called *generative*. Generative representations are always indirect. *Non-generative* representations can not reuse elements of the encoding. They can be either direct or indirect. Generative representations offer several advantages when compared to non-generative ones. Their ability to reuse elements of an encoded design improves the search efficiency in large design spaces as well as scalability by capturing design dependencies (Hornby 2003).

Depending on the nature of a relationship between the elements of a genotype and the elements of a generated phenotype, generative representations can be further classified as *implicit* or *explicit*. Implicit representations consist of a set of simple rules (e.g. cellular automata) that implicitly specify a design property, e.g. its shape, through an iterative construction process. Explicit representations are like procedural programs for constructing designs in an explicit manner.

Recently, there have been several attempts to coevolve representations of engineering systems during the evolutionary processes. This corresponds to a process in which a learning system adapts its own representation. De Jong and Oates (2002) proposed a coevolutionary approach to representation development where building blocks and their assemblies are coevolved. Also, Gero and Schnier (1995) worked on the evolution of the design knowledge representation, using genetic algorithms, in the context of case–based design. Such evolution is often necessary to produce inventive designs.

Traditional Design Representations

The majority of evolutionary design applications in structural engineering reported in the literature used relatively straightforward representations consisting of either binary strings or real-valued vectors. Thus, it is important to be aware of the strengths and weaknesses of both common approaches to represent engineering systems.

Binary representations are standard representations for canonical GA. The most straightforward and at the same time most common approach involves binary strings of fixed length. This type of representation is best suited for problem domains where solutions can be naturally represented as binary vectors, e.g. in some combinatorial optimization problems. In engineering design this type of representations has been widely used in structural topology optimization, e.g. in the ground structure approach (Dorn et al. 1964).

When a problem domain cannot be defined in terms of binary vectors, then a mapping from the binary space (genotypic space) to the domain space (phenotypic space) is necessary. Using this principle, binary string representations have been applied to continuous parameter optimization problems (Michalewicz 1996). In this case, a mapping between binary strings and real-valued parameters had to be specified. This approach has been widely used in many engineering design applications. Its advantage is that the standard GA operators (e.g. the bit-flip mutation, and one-, or two-point crossover) can be used. There are, however, some important drawbacks of this approach, too. Michalewicz (1996) argues that it is not appropriate because the problem space the GA is operating in is fundamentally different than that of the originally defined problem. Thus, search and optimization are conducted in a different space than the original one. Hence, the optimal results obtained in the binary search space might in fact not be optimal for the original problem. The genotype-phenotype mapping also introduces some additional nonlinearity to the objective function, and hence it may happen that the modified problem is more difficult to solve than the original one. Bäck (1996) points out another serious drawback of mappings from continuous to binary spaces. The mappings impose some granularity (resolution) and hence not all the points in the original continuous space can be expressed as binary vectors. So, it is possible that the optimal solution will not be found simply because it is not represented in the binary search space.

Another important problem with binary representations is related to the fact that one of the five major requirements on genotype-phenotype mappings, namely causality or continuity requirement, does not hold. In other words small changes in the binary space correspond to large changes in the real-valued parameter values and vice-versa. A frequently employed solution in this case is to use Gray encoding scheme (Bäck 1996).

Real-valued representation spaces have been traditionally used by ES researchers to solve complex continuous parameter optimization problems. Historically, they have been applied to engineering design problems, specifically to various fine tuned optimization problems. In ES, real-valued representations have traditionally been used as phenotypic representations, where no mapping between a genotype and a phenotype is necessary. Thus, the drawbacks associated with the mappings are eliminated in this case. There are, however, two major problems with real-valued representations which are somehow related. First, real-valued encodings allow for representation of only very specific problem domains, and that usually corresponds to fine-tuned optimization problems. As such, they are not applicable for creative design problems as I discussed it earlier. The second problem is that not every design problem can be expressed as a real-valued vector. There are many design problems, conceptual design problems being a good example of, that involve some symbolic or qualitative variables which cannot be encoded as real-valued parameters.

As stated earlier, representations are one of the three key elements in a successful implementation of evolutionary design. Throughout the years, enormous amount of experimental work has been devoted to studying various types of evolutionary representations. Despite this fact, very little is known theoretically about their influence on the performance of an EA. Initial framework for evolutionary representation theory has been recently proposed by Rothlauf (2002), but it is just the beginning of research on this important topic in EC.

2.1.4. Constraint-Handling Methods in Evolutionary Design

The vast majority of engineering design problems involves constraints of some kind. Thus, appropriate methods of handling constraints are extremely important for any optimization/search mechanism exploring designs spaces. Evolutionary algorithms, on the other hand, are unconstrained optimization procedures and hence it is necessary to somehow incorporate constraints into them. This section reviews the SOTA in constraint-handling methods in the context of evolutionary design. It also provides references to actual applications in structural engineering.

Coello Coello (2002) classifies constraint-handling methods used with EA into the following five major groups:

1. Penalty functions
2. Special representations and operators
3. Repair algorithms
4. Separation of objectives and constraints
5. Hybrid methods

Penalty Functions

Penalty functions have traditionally been the most common way of handling constraints incorporated in EA (Goldberg 1989; Michalewicz 1995). This method was initially proposed in the early 1940's in the context of traditional mathematical optimization by Courant (1943) and later extended by the operation research (OR) community in the 1960's (Caroll 1961; Fiacco and McCormick 1968). In the 1980's, penalty functions have been adopted by EC researchers to solve constrained optimization problems (Goldberg 1989; Goldberg and Samtani 1986) and since then have become the most popular, albeit not best as it has been shown in several studies (Richardson et al. 1989), method of handling constraints. Penalty functions effectively transform a constrained design problem into an unconstrained one by augmenting the objective function with a penalty term whose value determines the amount of constraint violation present in a particular solution (Coello Coello 2002). Contrary to classical optimization methods which use penalty functions of two kinds (i.e. exterior and interior), evolutionary design focused almost exclusively on exterior penalty functions because they do not require initial feasible solution to start with.

Various types of penalty functions have been proposed and studied. A general classification of the most commonly used types of penalty functions is presented below (Coello Coello 2002):

1. *Static penalty functions* which remain constant during an entire evolutionary process (Carlson 1995; Goldberg and Samtani 1986).
2. *Dynamic penalty functions* which change throughout an evolutionary run (usually increase over time) (Joines and Houck 1994).
3. *Annealing penalty functions* which use techniques based on simulated annealing (Michalewicz and Attia 1994).
4. *Adaptive penalty functions* which change according to feedback received from the search process (Bean and Hadj-Alouane 1992; Hadj-Alouane and Bean 1997; Nanakorn and Meesomklin 2001; Rasheed 1998; Smith and Tate 1993).
5. *Coevolutionary penalty functions* in which solutions are evolved in one population and penalty factors evolve in another population (Coello Coello 2000d).
6. *Death penalty functions* which immediately reject infeasible solutions (Schwefel 1981).

One of the major challenges in any application of penalty functions concerns achieving an appropriate balance of the penalty value. Large penalty values discourage EAs from exploring infeasible regions and the search is quickly moved inside the feasible region. On the other hand, low penalty values do not prohibit EA from searching infeasible regions most of the time. As a result of these findings, several EC researchers proposed the 'minimum penalty rule' which states that "penalty should be kept as low as possible, just above the limit below which infeasible solutions are optimal" (Coello Coello 2002). The problem with this formulation, especially for structural design applications, is that usually the constraints are not expressed in an algebraic form but instead as outcomes produced by structural analysis packages. Hence, an exact location of the boundaries between feasible and infeasible regions cannot be specified.

Methods of designing/configuring penalty functions for EC applications have been studied by Richardson et al. (1989). They offer several guidelines/heuristics that can be used to make evolutionary search in constrained design spaces more efficient:

- "Penalties which are functions of the distance from the feasible region are better than those which are merely functions of the number of violated constraints.
- For a problem having few constraints, and few solutions, penalties which are solely functions of the number of violated constraints are not likely to find solutions
- Good penalty functions can be constructed from two quantities, the maximum distance and the expected distance to the feasible region.

- Penalties should be close to the expected distance to the feasible region, but should not frequently fall below it. The more accurate the penalty, the better the solutions will be found. When penalty often underestimates this distance, then the search may not find a solution."

A number of applications showed, however, that there are many difficulties associated with penalty functions (Richardson et al. 1989), including, for example, a problem of defining good penalty factors. Thus, over the years, alternative approaches to handling constraints have been proposed by EC researchers.

Other Methods

Alternative attempts to handle constraints in evolutionary design include the development of *special representations* that simplify the shape of the search space and *special genetic operators* that preserve feasibility of generated solutions during the evolutionary run. Examples of applications of these methods include Bean's (1994) 'random keys encodings', Davidor's (1989) 'analogous crossover,' Michalewicz's (1996) GENOCOP, and Kowalczyk's (1997) constraint consistent GAs. Schoenauer and Michalewicz (1996) proposed a method that restricts the search to the boundary of a feasible region. It is based on a heuristic that in many cases the global solution lies on the boundary of a feasible region. In this method, the search mechanism crosses the feasibility boundary back and forth and special genetic operators are used to restrict the variation to the boundary of the feasible region (Schoenauer and Michalewicz 1997). The last set of methods in this category uses decoders (Michalewicz 2000a). In this case, chromosomes encode instructions on how to construct feasible solutions (Koziel and Michalewicz 1999). Each decoder imposes a mapping between a feasible solution and a decoded solution (Kim 1998; Koziel and Michalewicz 1999). Koziel and Michalewicz (1999) reported that decoders provided much better results than any other constraint-handling method on a representative set of test problems. They seem to be a very promising area of research in structural design because they can be used with problems of any dimensionality and do not require the objective function given in an algebraic form (Coello Coello 2002).

Repair algorithms are particularly well-suited for combinatorial optimization problems (Michalewicz 2000b). They are particularly efficient when the cost of transformation of an infeasible solution into a feasible one is low (Coello Coello 2002). They have been applied to many optimization problems (Liepins and Vose 1990; Michalewicz and Nazhiyath 1995; Mühlenbein 1992; Tate and Smith 1995). An interesting aspect of repair algorithms is whether, or not, a repaired individual should replace the original infeasible individual in the population. The spectrum of possible choices ranges from no replacement (repaired individuals are used only for evaluation and the original individuals remain in the population) (Liepins and Potter 1991; Liepins and Vose 1990) to the full replacement (all infeasible individuals are replaced with the repaired ones) (Nakano and Yamada 1991). Also, some intermediate approaches have been suggested where original infeasible solutions are replaced with some probability by the repaired solutions (Orvosh and Davis 1994). In structural design, repair algorithms have been used e.g. in (Kicinger 2004) to repair design concepts of steel structural systems in tall buildings not satisfying the symmetry requirement.

Another group of constraint-handling techniques can be broadly categorized as methods based on *separation of constraints and objectives* (Coello Coello 2002). Most representative techniques in this category include:

1. *Competitive coevolution* in which potential solutions (possibly infeasible) are evolved in one population and constraints are contained (but not evolved) in another population (Paredis 1994). Individuals representing potential solutions have high fitness when they satisfy a large number of constraints from the other population. On the other hand, an individual representing a constraint has high fitness if this constraint is violated by many potential solutions.
2. *Superiority of feasible points* which assumes that all feasible solutions are better than infeasible ones (Deb 2000; Powell and Skolnick 1993).
3. *Behavioral memory* that uses a special technique of ordering constraints in which the algorithm proceeds by sequentially satisfying the constraints imposed on the problem (Schoenauer and Xanthakis 1993).
4. *Multiobjective optimization methods* in which an original single-objective problem is transformed into a multiobjective one by treating all constraints in the original problem as objectives in the transformed problem (Coello Coello 2000a; Coello Coello 2000b; Parmee and Purchase 1994; Surry and Radcliffe 1997; Surry et al. 1995).

Finally, the last category of constraint-handling methods includes *hybrid methods* in which EAs are combined with other methods to solve constrained problems. In this category, several interesting methods were proposed, including:

1. *Lagrangian multipliers* in which a hybrid EA is formed by integration of a penalty function with mathematical programming methods including the primal-dual method and an augmented Lagrangian function (Adeli and Cheng 1994) that guarantees the generation of feasible solutions during the search (Kim and Myung 1997; Myung et al. 1995).

2. *Fuzzy logic* in which an EA is combined with fuzzy logic. In this method the original constraints are replaced by fuzzy constraints to allow a higher degree of tolerance for violating constraints that may occur close to the boundary of the feasible region (Le 1995; Le 1996).

3. *Immune system* models which have been initially proposed to maintain diversity in multi-modal optimization problems (Forrest and Perelson 1990; Smith et al. 1993) and later extended to solve constrained optimization problems (Hajela and Lee 1995b; Hajela and Lee 1996; Yoo and Hajela 1999).

4. *Cultural algorithms* which have been initially used to model cultural evolution (Reynolds 1994) and later applied to numerical optimization problems involving constraints (Chung and Reynolds 1996; Reynolds et al. 1995).

5. *Ant colony algorithms* inspired by colonies of real ants and initially proposed for solving combinatorial optimization problems (Colorni et al. 1991; Colorni et al. 1992) and subsequently extended to constrained optimization problems (Bilchev and Parmee 1995; Bilchev and Parmee 1996)

Excellent state-of-the-art reviews presenting theoretical and practical aspects of constraint-handling methods in evolutionary computation can be found in (Coello Coello 1999; Coello Coello 2002; Dasgupta and Michalewicz 1997; Michalewicz 1995; Michalewicz and Schoenauer 1996).

2.1.5. Multiobjective Evolutionary Design

Evolutionary multiobjective optimization (EMOO) is one of the most active research subfields within the EC community nowadays. EMOO methods are also highly relevant to engineering design problems because they were designed to handle multiple conflicting objectives which usually occur in real-world design problems. This section introduces the SOTA in evolutionary multiobjective optimization and presents recent developments in applications of these techniques to structural design problems.

There are two major goals of multiobjective optimization. First, one wants to find a large number of Pareto-optimal (Pareto 1896) solutions to a given problem. Second, the solutions to the problem should be widely differentiated (Deb 1999). Classical search and optimization methods (like weighted sum method (Chankong and Haimes 1983) or ε-constraint method (Haimes et al. 1971)) are not efficient for multiobjective problems because most of them cannot find multiple solutions in a single run, and even multiple runs do not guarantee finding different optimal solutions. On the other hand, EAs are well-suited to solve these kinds of problems because they are population-based and this property allows them to find an entire set of Pareto-optimal solutions in a single run. Additionally, they are significantly more robust, compared to the classical methods, particularly when issues like the shape or continuity of the Pareto front are a matter of concern (Coello Coello 2000c).

Initial research on using evolutionary methods for solving multiobjective problems was conducted by Rosenberg (1967). He suggested, but did not implement, a genetic search method involving multiple biochemical properties and objectives of a population of single-celled organisms. The first actual implementation was conducted by Schaffer (1984). In his dissertation, he proposed and successfully applied the vector evaluated genetic algorithm (VEGA) to multiclass pattern discrimination tasks in machine learning. Next significant progress in the field came with Goldberg's non-dominated sorting procedure outlined in (Goldberg 1989). Since that time, many researchers have developed various versions of multiobjective optimization algorithms. The most popular approaches reported in the literature include (Coello Coello 2000c; Deb 1999; Deb 2001):

1. *Aggregating functions* in which multiple objectives are combined into a single one using addition, multiplication, or any other combination of arithmetic operations (Syswerda and Palmucci 1991). Frequently, the weighted sum approach is adopted in which the objectives are multiplied by weighting coefficients representing the relative importance of the objectives (Jakob et al. 1992; Yang and Gen 1994). The major drawbacks of this method include difficulties in determining the appropriate weights and the fact that improper Pareto solutions may be generated in the presence of non-convex search spaces regardless of the weights used (Coello Coello 2000c).
2. *Vector evaluated genetic algorithm (VEGA)* proposed by Schaffer (1985). It handles multiple objectives by modifying the survival selection mechanism of the simple GA. Several variations of the original VEGA have been proposed and applied to various problems, including a groundwater pollution containment problem (Ritzel et al. 1994), and conceptual design of airframes (Cvetkovic et al. 1998).
3. *Target vector approaches* in which targets or goals have to be defined by a decision maker for each objective (Coello Coello 2000c). This group of approaches includes goal programming (Charnes and Cooper 1961), goal attainment (Chen and Liu 1994), and min-max approach. This last method, the weighted min-max, has been used by Haleja and Lin (1992) to optimize a 10-bar plane truss in which weight and

displacement were to be minimized, and by Coello Coello and Christiansen to optimize I-beams (1998) and truss designs (2000).

4. *Multiobjective genetic algorithm (MOGA)* proposed by Fonseca and Fleming (1993). It defines a rank of an individual based on the number of individuals in the current population by which it is dominated. MOGA has been used in many engineering design applications including for example a gas turbine controller (Chipperfield and Fleming 1995) and supersonic wings (Obayashi 1998; Obayashi 2002). Grierson and Khajehpour applied a variation of MOGA (called MGA) to conceptual design of office buildings (2002).
5. *Non-dominated sorting genetic algorithm (NSGA)* defined by Srinivas and Deb (1994) and based on Goldberg's (1989) notion of non-dominated sorting with a niche and speciation method. An improved version of this algorithm, called NSGA-II (Deb et al. 2000), equipped with elitisms and parameter-free sharing approach has been recently applied to a topological optimum design problem by Hamda et al. (2002b). In their approach, both the mass and the maximum displacement of a cantilever plate were minimized. Deb and Goel (2001) used a hybrid approach, NSGA-II and a hill climber, to solve several engineering shape optimization problems.
6. *Niched Pareto genetic algorithm (NPGA)* proposed by Horn and Nafpliotis (1993). It uses a tournament selection scheme based on Pareto dominance.
7. *Strength Pareto evolutionary algorithm (SPEA)* proposed by Zitzler and Thiele (1998) which integrates ideas from various existing evolutionary multiobjective optimization methods and adds some new elements to the evolutionary multiobjective algorithm.

Comprehensive surveys of various evolutionary multiobjective optimization methods, including detailed discussion on their strengths and weaknesses, can be found in (Coello Coello 1999; Coello Coello 2000c; Coello Coello et al. 2002; Deb 1999; Deb 2001; Van Veldhuizen and Lamont 1998).

2.1.6. Coevolutionary Design

Another important branch in evolutionary computation research that has recently received significant research attention is coevolution. I refer to *coevolution* as a phenomenon occurring when two or more populations (some researchers also include in this category single population models) simultaneously evolve and where no objective fitness function exists but rather individual's fitness is a subjective function of its interactions with individuals from coevolving populations (Rosin and Belew 1996; Wiegand 2003). Biological coevolution encountered in many natural processes has been an inspiration for a class of coevolutionary algorithms. Initial ideas of modeling coevolutionary behavior were formulated by Maynard Smith (1982) and Axelrod (1984; 1987). The competitive approach to coevolution has been since widely used in many game-theoretic models that arise in various disciplines, including economics, decision sciences, social sciences, etc. Initial ideas were further extended by Hillis (1991), Paredis (1994; 1995), and others and resulted in a new optimization procedure called coevolutionary genetic algorithm (CGA). Competitive coevolutionary models are especially suitable for problem domains where it is difficult to explicitly formulate an objective fitness function, for example in AI game-playing strategies, etc. Paredis (1994) applied competitive coevolutionary algorithms to constrained optimization problems. Recently, they have been used e.g. to coevolve cellular automata and the training cases for the majority classification problem (Pagie and Mitchell 2002).

Potter and De Jong (1994) proposed another approach to coevolution, namely a cooperative coevolutionary model. The motivation for this model comes from problem domains where explicit notions of modularity have to be introduced (Potter 1997). This model also provides appropriate framework for evolving solutions in the form of co-adapted subcomponents, and hence is of crucial importance for many engineering design problems. Usually, complex engineering design problems are decomposed into simpler problems and solved independently. This works fine for problem domains where the principle of superposition can be applied, i.e. for problems that can be linearly decomposed. That is no longer the case, however, for complex designs where nonlinear interactions take place among the subcomponents and make interacting members highly dependent on one another. For these domains cooperative coevolutionary model is more suitable because it allows for an explicit subcomponent coadaptation. Potter and De Jong (2000) proposed a cooperative coevolution architecture for evolving coadapted subcomponents and defined cooperative coevolutionary evolutionary algorithm (CCEA). This architecture has been subsequently analyzed from the evolutionary dynamics perspective (Luke and Wiegand 2002; Wiegand 2003) as well as from the perspective of collaboration methods that have been used (Wiegand 2003; Wiegand et al. 2001).

In general, coevolutionary design processes can be defined by 7 major attributes shown in Table 3. They describing ways in which coevolutionary systems can be set up (Wiegand 2003). The attributes include the payoff quality, methods of fitness assignment, methods of interaction, update timing, problem decomposition, spatial topology, and population structure.

Table 3. Attributes describing coevolutionary architectures

Attribute	*Attribute value(s)*		
Payoff quality	Cooperative	Competitive	Non-competitive
Methods of fitness assignment	Implicit	Explicit	
Methods of interaction	Sample size	Selective bias	Credit assignment
Update timing	Sequential	Parallel	
Problem decomposition	Partitioning methods	Temporal decomposition	
Spatial topology	Spatial embedding	Non-spatial embedding	
Population structure	Single	Multiple	

Coevolutionary models have been applied to several engineering design problems, particularly in architectural design. Maher and Poon (1996) suggested that it is often the case in a design process that requirements are reconsidered when a design solution is offered. Maher (1994) introduced the idea of coevolutionary design, where requirements and solutions evolve separately. Maher and co-workers (Maher and Poon 1995; Maher and Poon 1996; Maher et al. 1996; Maher and Wu 1998; Poon and Maher 1996a; Poon and Maher 1996b; Poon and Maher 1997) have been working on coevolutionary design in which two interrelated evolutionary processes occur. The first one is the evolution of design solutions while the second one is the evolution of requirements. In this case, the fitness function evolves with the requirements and it is different (local) at various stages of the coevolutionary design process. Also, the fitness function is used to identify the surviving solutions, but its convergence simply means that there is no progress in the evolution since no new and better solutions are being produced.

The only work known at this time which uses cooperative coevolutionary algorithms in structural optimization was conducted by Nair and Keane (2002). They used CCEA to optimize cross-sections of members of planar truss systems (single objective weight minimization problem). The optimized truss systems were decomposed and coevolved in separate populations.

2.1.7. Evolutionary Computation in Structural Engineering

The history of evolutionary computation in structural engineering can be traced back to the mid 1970's and early 1980's (Goldberg and Samtani 1986; Hoeffler et al. 1973; Lawo and Thierauf 1982). The vast majority, if not all, of early papers discussing EC applied to structural engineering were focused on structural optimization problems. Strong emphasis on various aspects of structural optimization remained the major focus of research in this field until now with relatively few exceptions which mostly addressed the issues of creativity in structural design and more sophisticated forms of representations of structural systems (Hamda et al. 2002a).

Emergence of EC in structural optimization was a consequence of encountered problems and deficiencies of formal methods, including mathematical programming and the optimality criteria method (Berke and Khot 1987), when applied to more complicated structural design domains. Formal structural optimization methods based on the assumption of continuity worked well on relatively well-formed problems in which the structural configuration of members was assumed and fixed during an optimization process while the task was to find the optimal sizing (dimensions) of members' satisfying at the same time imposed design requirements and constraints. The simple generalization of this problem by allowing variations of a system's configuration greatly increased the complexity of the optimization task and rendered many traditional methods inadequate. This issue became a starting point for a development of two major approaches to structural optimization that exist today: enhanced formal methods and heuristic methods.

Structural Design Problems

The problems addressed by structural optimization can be divided into three major categories:

- Topology (layout) optimization also known as topological optimum design (TOD) – looking for an optimal material layout of an engineering system
- Shape optimization (SO) – seeking optimal contour, or shape, of a structural system whose topology is fixed
- Sizing optimization – searching for optimal cross-sections, or dimensions, of elements of a structural system whose topology and shape is fixed

A structural design problem in each of the categories can be further classified as a continuum or discrete optimization problem. Figure 3, a modified version of a figure presented in (Jakiela et al. 2000), shows the three categories of structural optimization for continuum design problems while Figure 4 shows the same categories for discrete problems.

The three categories are closely related to three major stages of engineering design process described earlier, i.e. TOD is conducted in the conceptual design stage, SO in the embodiment design stage, and finally sizing optimization is performed in the detailed design stage. As stated earlier, the three categories of structural optimization problems have been addressed by both formal optimization methods and heuristic methods.

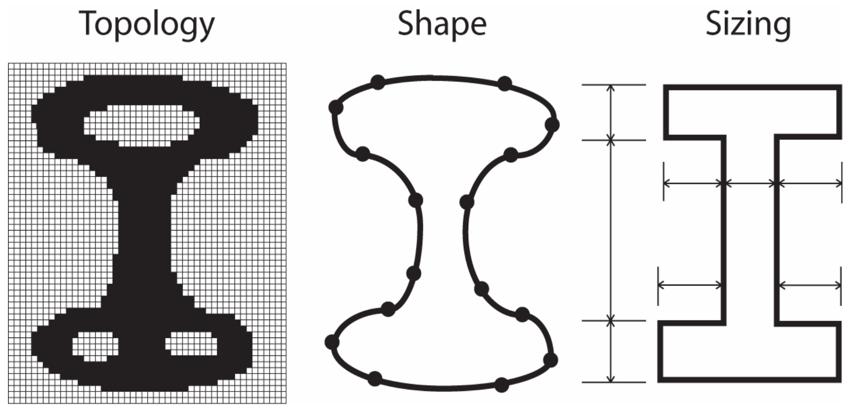


Figure 3. Topology, shape, and sizing optimization for continuum structural design problems

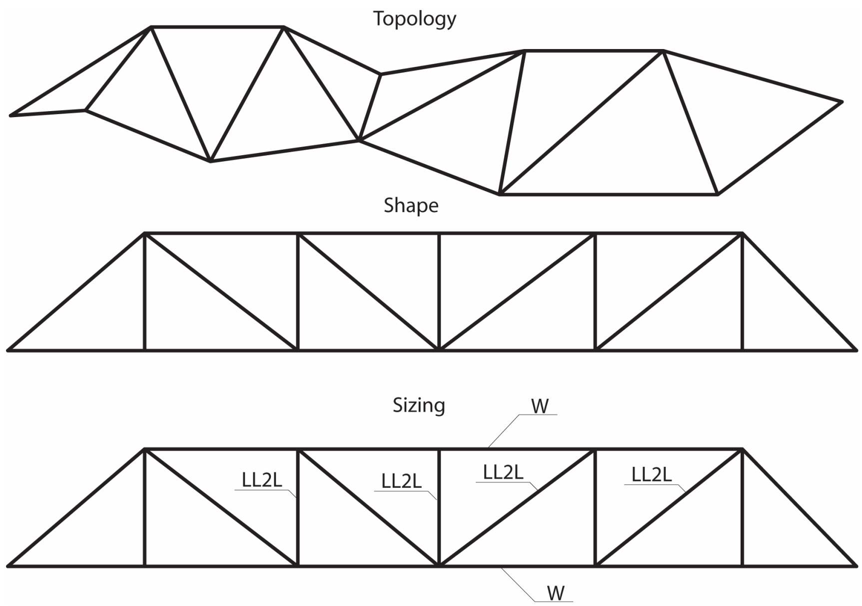


Figure 4. Topology, shape, and sizing optimization for discrete structural design problems

Formal methods have been most successful when applied to sizing optimization problems which are usually well-defined in terms of mathematical models. Mathematical programming methods (Schmit 1981) and optimality criteria method (Berke and Khot 1987) have been efficiently applied to solve these problems. Heuristic methods, including EAs, have also been applied to structural sizing problems (Lin and Hajela 1993; Schoenauer and Wu 1993). On the other hand, TOD problems, located on the other end of the structural complexity spectrum, have been most successfully approached using heuristic methods, including simulated annealing (Anagnostou et al. 1992) and EAs (Chapman et al. 1994; Hajela and Lee 1995a; Jensen 1992; Kane and Schoenauer 1996). Structural shape optimization has been a kind of middle ground where both formal and heuristic methods are used and complement one another.

Topological Optimum Design

TOD has been an area of significant research efforts for the last forty years. Initial investigations in the late 1970's and early 1980's were conducted using formal methods. Generally, TOD problems can be divided into two major groups: continuum TOD and discrete TOD. In the continuum TOD, the design domain is discretized into small, rectangular elements (rectangular grid) where each element contains material or void. Formal methods addressing this problem include the homogenization method (Bendsoe and Kikuchi 1988) in which each element in a grid contains composite material of continuously-variable density in [0,1] and orientation. Xie and Steven (1992) proposed evolutionary structural optimization (ESO) method which follows the concept of removing lightly stressed elements. The name of this method is confusing because the method is not based on EC principles but rather evolution is understood in a more general context as a process of gradual removal of inefficient material from a structure. The EC approach to the continuum TOD problem based on GAs has been developed by Sandgren et al. (1990) and Jensen (1992). In their approach, a GA determines the optimal layout of material and void in a cantilever plate (represented as a bit array) such that the structure's weight is minimized subject to displacement and/or stress constraints. This work has been subsequently extended by Chapman et al. (1994) to optimize finely-discretized design domains and to obtain families of highly fit designs. Recently, more advanced forms of representations for continuum TOD problems have been proposed, including Voronoi-based representations (Periaux and Winter 1995; Schoenauer 1996), which are based on concepts of Voronoi diagrams studied in computational geometry, and IFS representations based on fractal theory (Hamda et al. 2002a). Also, Hamda et al. (2002b) considered a continuum TOD as an evolutionary multiobjective optimization problem.

Discrete TOD problems consist in determining the optimal element connectivity from a finite, albeit large, number of possible connections (Topping 1983). Two major problem domains addressed in early research in this area include truss structures and frame structures. An initial problem formulation in the context of linear programming using the ground structure approach was proposed by Dorn et al. (1964). While traditional linear programming methods proved to be successful in finding optimal topologies for small problems, they were rendered inadequate when the size of the problems considered was scaled up (increase in the number of design variables or the number of grid points in the ground structure approach). The discontinuous nature of this design problem was another reason for inefficiency of formal methods. Initial applications of GAs to optimize topology of discrete-member trusses were conducted by Shankar and Hajela (1991), Hajela et al. (1993), Grierson and Pak (1993a), and Hajela and Lee (1995a). Bramlette and Bouchard (1991) used EC to three-dimensional structures in the context of aircraft design. Koumousis & Georgiou (1994) applied GAs to the topology optimization of steel truss structures. Bohnenberger et al. (1995) applied GAs to optimize topologies of truss structures in pylons. Rajan (1995) applied GAs to optimize topology, shape and member sizing of truss structures. Nakanishi and Nakagiri (1996; 1997) used GAs to solve 2D topology optimization problems for both frames and panel structures. Rajeev and Krishnamoorthy (1997) used variable-length string representations to optimize truss structures. Murawski et al. (2001) and Kicinger et al. (2004) applied ES to optimize topology of steel structural systems in tall buildings. Soh and Yang (2001) introduced a GP-based approach to TOD of truss structures. In a subsequent work (2002), they proposed a GP-based methodology for the automated optimum design of structures. Recently, Azid et al. (2002) applied a GA with real-valued representations to optimize topologies of three-dimensional trusses.

SOTA reviews of current research in formal methods for TOD problems can be found in Rozvany et al. (1995), Bendsoe and Sigmund (2002), and Xie and Steven (1997) whereas recent research developments in applications of EC to TOD problems can be found in (Hajela and Vittal 2000).

Shape Optimization

Shape optimization maintains a fixed topology of structural designs but changes their shape or node locations. Similar to the TOD case, shape optimization problems can be divided into two major groups: continuum SO and discrete SO. Continuum SO addresses shape optimization problems in the context of 2D or 3D continuum structures. Traditionally, in continuum SO, "a shape is defined by the oriented boundary curves [2D structures] or boundary surfaces [3D structures] of the body … and the optimal form of these boundaries is computed" (Bendsoe and Sigmund 2002). Formal methods for solving continuum SO problems are well-established and extensive literature is available (Bennet and Botkin 1986; Haslinger and Neittaanmaki 1996; Pironneau 1984). Sensitivity analysis for shape optimization problems is discussed in (Sokolowski and Zolesio 1992) and application of the homogenization method to this problem is offered in (Allaire et al. 1997). ESO, introduced earlier, has also been used to shape optimization (Xie and Steven 1992). Evolutionary computation methods have also been applied to solve continuum SO problems. Research on shape optimization of structural members has been conducted by Jenkins (1991a; 1991b), Richards and Sheppard (1992), and Watabe and Okino (1993). Kita and Tanie (1998; 1999) and Annicchiarico and Cerrolaza (1999; Cerrolaza and Annicchiarico 1999) used GAs to optimize the shape of continuum 2D structures through B-spline functions. A GA was used to find optimal locations of knots of B-spline functions. Wibowo and Besari (1998) applied GAs to optimize shapes of oval axially symmetric shells. Annicchiarico and Cerrolaza (2001) applied GAs to shape optimization of 3D finite element models. Woon et al. (2001) investigated alternative encodings of GAs for continuum SO using the actual coordinates of boundary nodes.

Discrete SO methods conduct shape optimization through variations in geometry of discrete truss and frame structures introduced through changes in locations of nodes (Pedersen 1987; Vanderplaats 1975). Various mathematical programming methods have been used to discrete SO problems, including linear, nonlinear, and dynamic programming (Topping 1983). In the case of shape optimization of truss structures, discrete TOD methods using the ground structure approach have been extended to include optimization of the nodal point locations for a given number and connectivity of nodal points (Bendsoe and Sigmund 2002). Initial applications of EC methods to discrete SO problems have been conducted by Grierson and Pak (1993a; 1993b) in the context of truss structures. Soh and Yang (1996) applied fuzzy controlled GAs to optimize the shape of planar and spatial truss structures. Bohnenberger et al. (1995) applied GAs to optimize shapes of truss structures in pylons. Keane and Brown (1996) used GAs to optimize the shape of a satellite boom with respect to its vibration performance.

SOTA reviews in traditional mathematical approaches to continuum shape optimization problems are presented in (Allaire and Henrot 2001; Kawohl et al. 2000). Recent developments in formal methods for discrete SO problems can be found in (Bendsoe and Sigmund 2002; Nishino and Duggal 1990). Recent developments in applications of GAs to design of steel structures are described in (Pezeshk 2002).

Sizing Optimization

Sizing optimization problems involve finding optimal cross-sections, or thicknesses, of elements of a structural system whose topology and shape is fixed. It is the easiest of the three structural optimization problems discussed earlier and relatively well-understood. Research on formal methods of solving these kinds of problems has a long history and extensive literature is available on the topic (Arora 1989). First applications of EC to structural optimization problems involved these types of optimization problems. Lawo and Thierauf (Lawo and Thierauf 1982) used ES to optimize members of a planar six-story frame subjected to earthquake loading. Goldberg and Samtani (1986) applied a GA to optimize cross-sections of members of a 10-bar plane truss. Hajela (1990; 1992) investigated cross-section optimization of discrete member trusses using GAs. Deb (1991) applied GAs to optimize designs of welded beams. Jenkins (1992) proposed a GA-based design environment to optimize plane frame structures. Rajeev and Krishnamoorthy (1997) applied GAs to optimize cross-sections of generalized trusses. Recently, Jarmai et al. (2003) applied genetic algorithms to design welded I-section frames and compared their performance with other nonlinear optimization algorithms operating in a constrained representation space.

Historical Perspective

A summary of major applications of EC in structural design since its beginning in the mid 1970's is provided in a chronological order in Appendix A. The applications are classified with respect to the application domain and major EC characteristics, including the representation type, the evolutionary algorithm used, the fitness function, and methods of handling constraints. A chronological classification of the EC applications in structural design clearly shows three major periods in the development of the field:

1. **Period of early explorations (1986-1995)**
   During this initial stage, simple evolutionary algorithms (mainly, if not exclusively ES and GAs, sometimes combined with other traditional optimization methods) were applied to relatively simple structural engineering problems (sizing optimization of simple 2D engineering systems). Researchers focused on using standard design representations, i.e. binary strings and real-valued vectors, single objective fitness functions (usually the minimization of weight), and fairly traditional constraint-handling methods involving various variations of the penalty functions (see section 2.1.4).
2. **Period of exploration & exploitation (1996-2000)**
   This period can be best characterized as a period of exploring alternative choices for various components of the evolutionary algorithms and improving the process of optimization of more complex design problems. Researchers explored various kinds of representations of engineering systems, including Voronoi-based representations and integer-based representations. Significant research efforts were also focused on tuning the genetic operators to particular problems, e.g. by adapting mutation and crossover rates during the evolutionary design processes. Initial exploration of alternative constraint-handling methods has also been conducted, including immune networks, behavioral memory, and fuzzy logic. Several multiobjective approaches to structural design problems have been reported as well.
3. **Period of rapid growth (2001-present)**
   Currently, evolutionary computation is a fully recognized structural optimization paradigm and is frequently used not only by researchers but also by practitioners. Nowadays, research efforts are focused on solving much more complex structural design problems and on studying more advanced evolutionary models, including parallel EA, multiobjective optimization, and variable-length representations, in the context of structural design. Also, initial exploration of the potential of using coevolutionary models is being conducted.

Summary

The field of evolutionary design and its applications to engineering design is far from maturity and continues to rapidly grow and develop in many exciting new directions. In this section, I summarize several of the most promising areas of new research. They can be grouped into the following five classes:

1. **Integrated structural design support tools**

   As the size and complexity of structural problems in the field of evolutionary design continues to increase, there are several scaling-up issues that need to be addressed, including computation time and parallel architectures. Computation time in evolutionary design mostly depends on the evaluation of the fitness of generated designs (frequently 90-95%, or more, of computation time). In the past, when computational costs were high, researchers developed a variety of techniques to minimize the computational effort. One of the most popular techniques involved separation of the stages of conceptual, preliminary, and detailed design, and developing separate tools for each stage (Arciszewski and De Jong 2001). Nowadays, however, the cost of computation continues to decrease and this trend is likely to persist in the future. Second, parallel computer architectures are now readily available. Considering also the fact that EA have a natural mapping onto parallel architectures, it is my belief that computational costs should not be the primary factor in developing new integrated evolutionary-based structural design support tools. These tools will treat all the stages of a design process as phases of a single integrated design process. Research efforts in this direction are led by Parmee and co-workers (Parmee 2001).

2. **Open-ended representations**

   An appropriate representation of an engineering system is one of the key issues in any structural design application. Today, it becomes even more important because the increased complexity of considered design problems raises some difficult internal EA issues on how to best represent and evolve complex designs (Arciszewski and De Jong 2001). Another motivation comes from the fact that there is an emerging trend to apply evolutionary design techniques not only to strictly optimization tasks but rather this technique is being gradually more and more useful in finding creative/novel design concepts. Both issues lead to open-ended representations which don't encode entire designs but rather rules on how to construct these designs (see section 2.1.3). Representations of this type are also inspired by the processes occurring in nature, where we observe evolution manipulating the genetic plans for complex objects rather than the objects themselves. The organisms are then built from the plans via a developmental process called morphogenesis.

3. **Alternative constraint-handling methods**

   Almost every structural design problem involves some kind of constraints. Up to very recently, various variations of penalty functions were virtually the only method of handling constraints. On the other hand, a number of applications showed that there are many difficulties associated with this approach when applied to highly constrained optimization problems. Studies focused on estimating a true potential of alternative constraint-handling methods (discussed in section 2.1.4) constitute another promising area for future research of vital importance to structural design.

4. **Multiobjective structural design**

   Structural design problems are inherently multiobjective and often involve a large number of conflicting criteria. So far, research in evolutionary structural design concentrated almost exclusively, with few notable exceptions, on single objective problems. At the same time, the field of evolutionary multiobjective optimization provides new and efficient methods, described in section 2.1.5, of solving these types of problems. Multiobjective structural design may become one of the most promising areas of research in structural design, particularly when not a single optimal design solution is sought but rather a set of alternative optimal designs.

5. **Coevolutionary structural design**

   Coevolutionary design is an emerging area of research with many unanswered questions. There is a lot to be done to understand the true potential of this paradigm in structural design. Initial findings coming from evolutionary computation community suggest that coevolutionary models might be particularly suitable for complex design spaces that can be relatively well decomposed and when the major goal is not the optimality of design solutions in a global sense, but rather their robustness (design for reliability) (Wiegand 2003). As stated earlier, very little has been done in this area and it is potentially one of the most promising paths of future research.

Research reported in this dissertation is mostly related to points 1 and 2 described above, i.e. building integrated design support tools and using open-ended representations in engineering design.

2.2. Overview of Cellular Automata

2.2.1. Cellular Automata

Cellular automata are one of the simplest mathematical representations of complex systems (Wolfram 1983). As such, they are useful idealizations of a dynamical behavior of various systems. They appear to capture many essential features of a complex self-organizing behavior observed in real world systems. CAs are prototypical models of complex systems and processes consisting of a large number of *identical*, *simple*, and *locally interacting* components. CAs can be used to study pattern formation and gain some insight into self-organization processes. The CAs research has generated great interest over the last forty years because of their ability to exhibit very complex patterns of behavior using a set of relatively simple underlying rules. Recently, Wolfram (2002) suggested that cellular automata and other simple programs may better model nature's most essential mechanisms than traditional mathematical equations.

The origins of CA research are commonly associated with two people: John von Neumann and Slanislaw Ulam. Von Neumann proposed CAs as a reductionist model for biological evolution (1951). Following suggestions by Ulam (1952; 1974), he used discrete rather than continuous dynamics to construct a two-dimensional self-replicating automaton. It was the first discrete parallel computational model formally shown to be a universal computer as defined by Turing (1936).

CAs have been successfully applied in physics, biology, chemistry, economy, geology, and other disciplines. Some specific examples of modeled phenomena include fluid and chemical turbulence (d'Humieres and Lallemand 1986; Gerhadrt and Schuster 1989), growth of crystals

(Kessler et al. 1990), DNA evolution, social dynamics (Axtell and Epstein 1996), patterns of electrical activity in neural networks (Franceschetti et al. 1992), discrete versions of partial differential equations in one or more spatial variables, path planning for mobile robots (Marchese 2002), etc. There have also been several engineering applications of cellular automata, including models of traffic flow and of transportation systems (Marinosson et al. 2002; Nagel 2002). Structural engineering applications of CAs are discussed in section 2.2.3.

Following Ilachinski (2001), we can distinguish 5 generic characteristics of CAs:

- *Discrete lattice of cells*: the system consists of usually 1-, 2-, or 3-dimensional lattice of cells (higher dimensional extensions are also possible but rarely used in practice).
- *Homogeneity*: all cells are equivalent (although there are also models using non-uniform CAs (Sipper 1997)).
- *Discrete states*: each of the cells can be in one of the finite number of possible discrete states.
- *Local interactions*: each cell interacts only with cells contained in its local neighborhood.
- *Discrete dynamics*: at each discrete time unit, each cell updates its current state according to a transition rule taking into account the states of cells in its neighborhood.

The simplest possible CAs, called elementary CAs, consist of a one-dimensional lattice of cells, in which each cell can be in one of two possible states. The value of each cell at a next time step is determined by a value of the cell itself and its two closest neighbors. In other words, an elementary CA is a one dimensional CA with binary states and with a local neighborhood of size 3 (or a neighborhood radius equal to 1). Results of a process of iteration of an elementary CA are presented in Figure 5a).

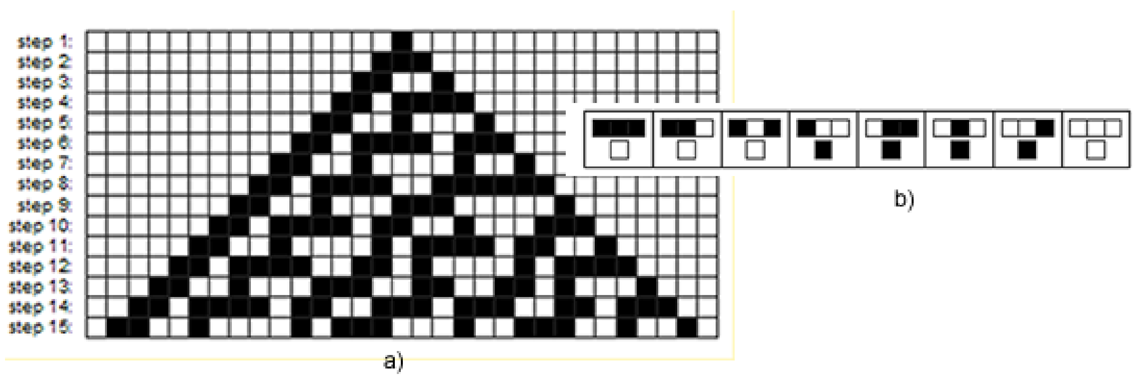


Figure 5. a) Process of iteration of an elementary CA and b) a transformation rule determining the values of cells at a next time step

The top row of cells (step 1 in Figure 5a)) is iterated 14 times (steps 2-15) using a CA transformation (or update) rule shown in Figure 5b). The CA transformation rule specifies all possible (8 in the case of an elementary CA) combinations of cell state values in a local neighborhood of size three (the top row) and the values achieved by the central cells at a next time step (bottom row). Increasing the number of cell state values or the size of the local neighborhood causes a rapid growth in the number of possible CA rules. For example, changing the number of cell state values to 3 with the same size of the local neighborhood yields 7,625,597,484,987 possible CA rules compared 256 CA rules for elementary CAs. There is, however, a way to significantly reduce it by introducing a concept of a totalistic CA. In a totalistic CA, a new value of each cell depends only on the *average* value of the neighboring cells and the cell itself, and not on their individual values (Wolfram 2002). For example, due to averaging, there are only 2187 possible totalistic CAs with 3 values and the neighborhood of size three compared to 7,625,597,484,987 rules found in the corresponding standard CAs.

Figure 6 shows a process of iteration of a totalistic CA with three state values (see Figure 6a)) and a totalistic CA transformation rule (see Figure 6b)). In this particular example, the rule specifies all 7 possible local neighborhoods of size three corresponding to 7 possible *average* cell state values, i.e. 0, 0.33, 0.66, 1, 1.33, 1.66, and 2. They are denoted graphically by various shades of gray (the top row). The values achieved by the central cells at a next time step, i.e. 0, 1, and 2 are shown in the bottom row.

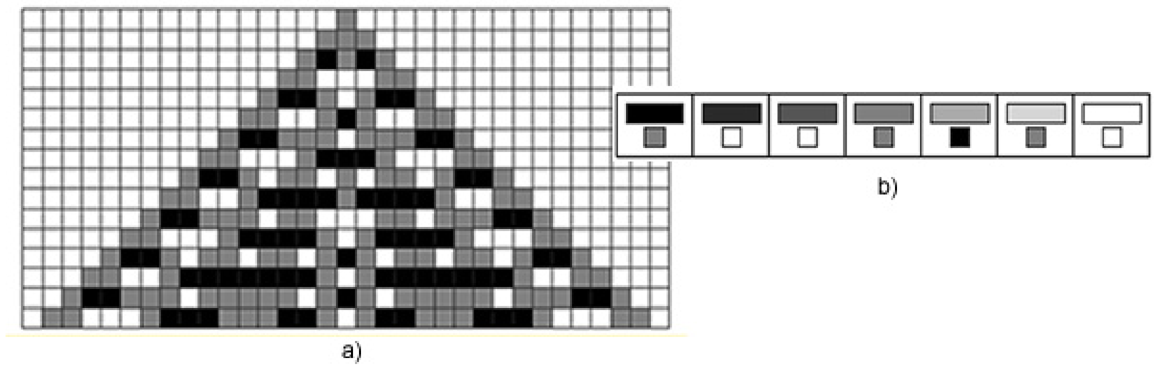


Figure 6. a) Process of iteration of a totalistic CA and b) its transformation rule

Formally, a one-dimensional CA (1D CA) can be defined in the following way. Let $c_i(t)$ denote the value of the *i*th cell at time *t*. A CA evolves according to a rule *F* that is a function of $c_i(t)$ and other cells that are within a neighborhood *r* of $c_i(t)$:

$$c_i(t+1) = F(c_{i-r}(t), c_{i-r+1}(t), ..., c_{i+r-1}(t), c_{i+r}(t))$$

Each cell can take on one of the *k* possible values, that is $c_i(t) \in \{0,1,...,k-1\}$. Thus, the rule *F* is completely defined by specifying the value assigned to each of the k^{2r+1} possible (*2r+1*)-tuple configurations for a given *r* neighborhood. Since *F* itself assigns any of *k* values to each of the k^{2r+1} possible (*2r+1*)-tuples, there are a total of $k^{k^{2r+1}}$ possible rules.

When looking at CAs from a dynamical systems perspective (see section 2.3), they can be treated as abstract discrete dynamical systems that produce inherently interesting, and potentially novel, behavioral patterns. As Wolfram (1983) has shown, all one-dimensional CAs evolving from random initial configurations generate patterns that can be classified into one of only four basic behavioral classes:

- Evolution leads to a homogenous state, in which all cells eventually attain the same value
- Evolution leads to either simple stable states or periodic and separated structures
- Evolution leads to chaotic nonperiodic patterns
- Evolution leads to complex, localized propagating structures

Figure 7 shows graphically the four classes of behavior defined above. Dynamical behavior of an elementary CA presented in Figure 7a) represents a homogeneous state behavior, i.e. a class one behavior in the taxonomy defined above. Figure 7b) presents periodic behavior in which the period's length is equal to 2 (a class two behavior). Figure 7c) shows much more complex behavior where no regularity and periodicity can be found (a class three behavior).

Finally, Figure 7d) shows a CA exhibiting a class four behavior with localized, propagating structures.

The first three behavioral patterns qualitatively resemble behavior observed in continuous systems (see section 2.3). Homogenous states are analogous to fixed-point attracting states, asymptotically periodic states are analogous to continuous limit cycles, and finally chaotic states are analogous to strange attractors. More complex structures occurring in systems exhibiting a class four behavior do not seem to have obvious analogues in continuous systems. Sometimes they are characterized as soliton-like structures in their appearance (Ilachinski 2001).

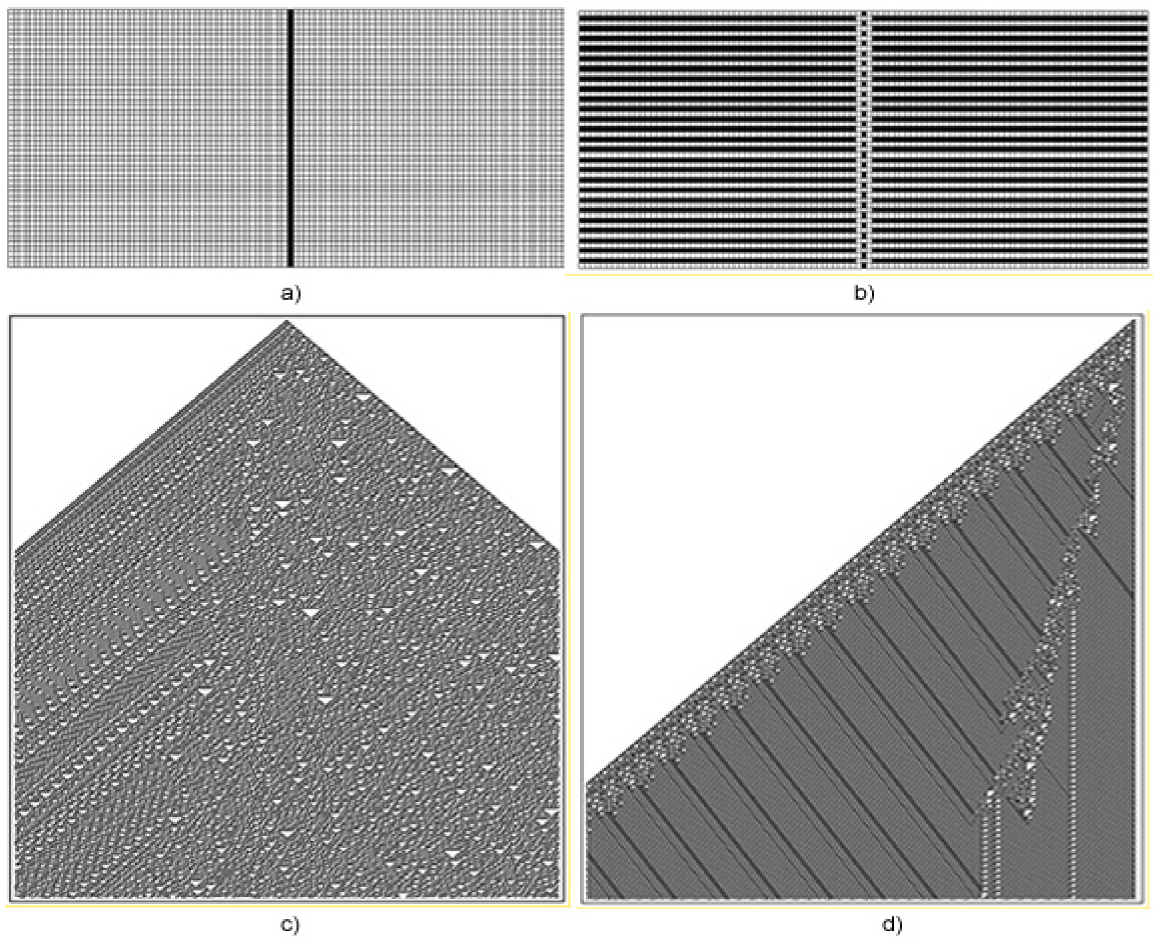


Figure 7. Four classes of dynamical behavior produced by elementary CAs

In the examples, even the complex, nonperiodic behavior shown in Figure 7c) was generated by the simplest possible cellular automata. Thus, an apparent complexity of behavior does not necessarily imply the complexity of mechanisms generating that behavior. Conversely, even the simple systems, like elementary cellular automata, can exhibit a very irregular and counter-intuitive behavior. This observation contributed to the development of a recently proposed New Kind of Science (Wolfram 2002).

Two-dimensional cellular automata (2D CAs) are generalizations of one-dimensional systems in which the lattice of cells is no longer one-dimensional but it is extended to two dimensions. 2D CAs can be defined using a set of parameters known from 1D CAs but with several additional properties. These additional properties include an initial configuration of cells which is now two-dimensional aw well as CA transformation rules that now have to take into account two-dimensional local neighborhoods of a current cell. In order to fully define a 2D CA transformation rule one not only has to specify a radius of the local neighborhood *r* but also its shape. Two most popular shapes of 2D local neighborhoods include von Neumann neighborhood (diamond-shaped neighborhood) and Moore neighborhood (square-shaped neighborhood). Figure 8 shows the impact of the shape and radius on a two-dimensional local neighborhood in a 2D CA.

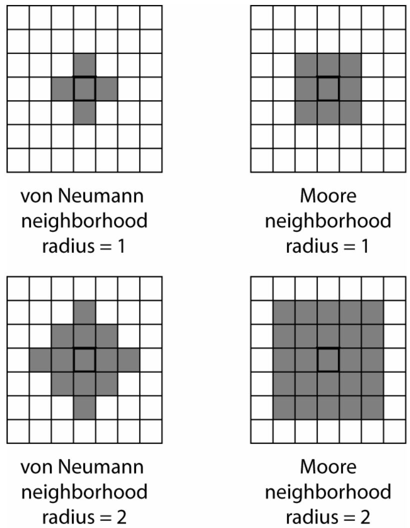


Figure 8. Impact of the shape and radius parameters on a local neighborhood in a 2D CA

As it was the case with 1D CAs, the 2D CA transformation rules can be also defined based on average values of the cells in the local neighborhood. In this way, so-called totalistc 2D CAs are defined. In fact, this type of 2D CA is more common due to the fact that the number of possible transformation rules (and hence the size of the search space) rapidly increases. Figure 9 shows several steps of a process of iteration of a totalistic 2D CA started with a 2D lattice of cells with a single cell with the state value equal to 1 (a single black cell in the middle of the 2D lattice shown at step 0).

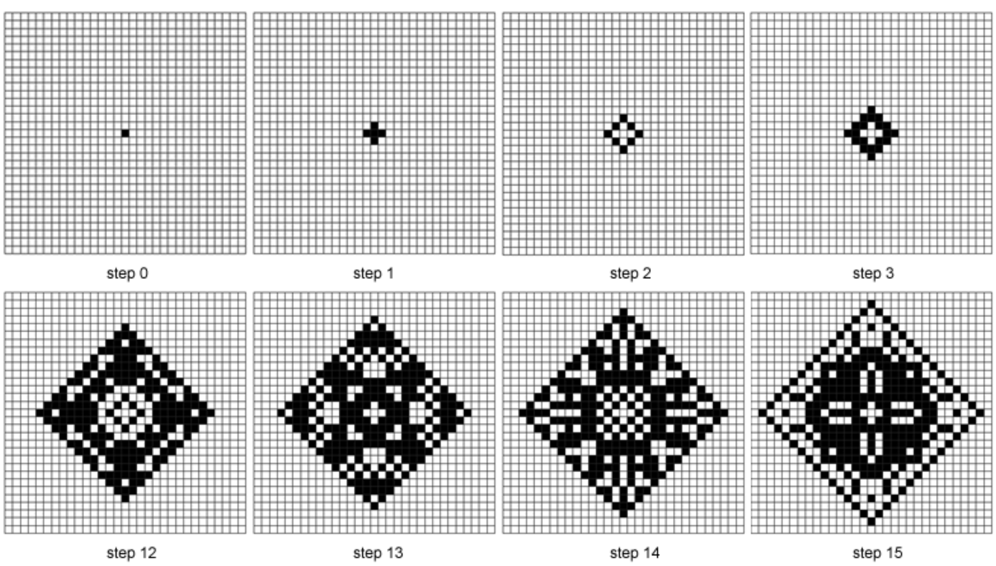


Figure 9. Several steps of iteration of a two-dimensional cellular automaton

2.2.2. Numbering Scheme for Cellular Automata Rules

The numbering scheme presented here and used in the remainder of this dissertation has been initially proposed by Wolfram (1983) for elementary CA rules. It can be, however, generalized to describe arbitrary CAs. A detailed description of the scheme is given below.

Figure 10 shows an example of a transformation rule of an elementary CA. Above, in the top row, all $k^{2r+1} = 2^3 = 8$ possible combinations of values of three variables (neighborhood of size three) are given. Below, in the bottom row, values achieved by the central cells at a next time step are given. Thus, **if we assume the same ordering of the local neighborhoods** as shown in the top row of Figure 10 then any elementary CA rule can be uniquely defined by a single eight-digit binary number. Each digit in this number specifies the value achieved by the central cell at a next time step for a given combinations of cells in a local neighborhood. This binary number can be also written as a decimal value, and this is what I call in this dissertation a CA transformation rule number, or simply a CA rule. An example of a CA rule shown in Figure 10 defines the rule 110. Any elementary CA can be uniquely defined by a rule number from 0 to 255. A graphical representation of the numbering scheme is presented in the middle of Figure 10. Here, a black square denotes 1 and a white square denotes 0.

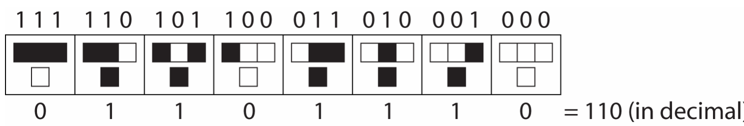


Figure 10. Numbering scheme for elementary CAs

As stated earlier, the numbering scheme can be generalized to describe an arbitrary type of a CA. For example, for a CA with 3 possible state values the only difference in the numbering scheme would involve a change in the base. In this case the base would not be binary but rather ternary. The value in a ternary base can be subsequently written as a decimal value, as I discussed it earlier. The scheme also works for totalistic CAs. Figure 11 shows an exemplary totalistic CA with three possible state values and with a local neighborhood of size three.

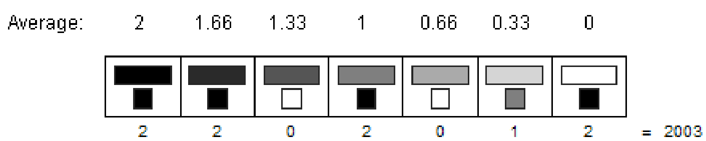


Figure 11. Numbering scheme for totalistic CAs

In this particular case, a totalistic CA can be uniquely defined by a seven-digit ternary number, which can be subsequently converted to a decimal value.

2.2.3. Cellular Automata in Structural Engineering

As discussed earlier, CAs have been a subject of significant research interests in various disciplines of science. There have also been several studies on using cellular automata in structural engineering. One of the first applications of CAs to shape optimization is due to Inou et al. (1994; 1998). They used local rules such as 'death', 'birth', and 'division' to investigate self-organization of topologies in structural systems. Kundu et al. (1997) applied CAs to optimize the shape of structural plates. Kita and Toyoda used CAs to optimize both the shape and the topologies of two-dimensional elastic structures (2000) as well as to optimize cross-sections in truss structures (2001). Hajela and Kim (2001) applied GAs to search the space of CA rules in structural analysis of 2D elastic structures.

2.3. Overview of Dynamical Systems, Chaos, and Complex Systems

2.3.1. Dynamical Systems and Chaos Theory

The beginnings of the field of dynamical systems and chaos theory are commonly associated with Henri Poincaré who in 1890's studied a simplified model of a solar system consisting of three bodies. Using his innovative methods of modeling dynamical systems (a qualitative approach) he discovered that even this very simplified model produced incredibly complicated behavior (1897). Poincaré's methods proved to be very useful for describing the behavior of a wide variety of physical systems. After Poincaré, other important contributions were made by Birkhoff (1927), Cartwright (Cartwright and Littlewood 1951), Kolgomorov (1958), and others. A fast progress in the science of complexity, however, was possible only with the advent of personal computers in 1960's and 1970's.

Lorenz (1963) published his famous work on deterministic nonperiodic flow occurring in his computer model of a global weather. He discovered a surprising *order in the middle of a chaotic motion*, now called Lorenz attractor. He was also able to identify another hallmark of chaos, namely the *sensitivity to initial conditions*. Scientists equipped with a modern research tool, i.e. a personal computer, have started numerical explorations of chaotic dynamics in almost all disciplines of science: mathematics, physics, biology, physiology, engineering, and many others. The underlying rules proved to be universal for all fields.

Formally, a *dynamical system* can be defined as a function $\varphi : \mathrm{T} \times \mathrm{M} \rightarrow \mathrm{M}$ such that the following properties hold:

1. $\varphi(0, x) = x$ for all $x \in \mathrm{M}$
2. $\varphi(t, \varphi(s, x)) = \varphi(t + s, x)$ for all $t, s \in \mathrm{T}, x \in \mathrm{M}$

where T is a time set, and M is a state space.

2.3.2. Complex Systems

A *complex system* is a dynamical system that consists of large number of mutually and typically nonlinearly interacting parts. The field of complex systems is, however, a relatively young discipline of science and, as such, not yet well defined. One of its distinguishing features is an *emergent behavior*, i.e. a type of global behavior of an entire system which exhibits some characteristics neither possessed by not directly derived from any of its parts (Ilachinski 2001). It is not enough to understand a complex system in terms of its components out of which it is constructed but one also has to include in the model both the topology of interconnections and interactions between these components.

Complex systems can be found on many levels in nature and society. On a micro level, they are found in spin systems as well as in reaction-diffusion systems which give rise to a complex spatio-temporal behavior. On macro scales, they are present at various societal levels, for example in economic and social systems where the agents interact, compete for resources, and cooperate. But, certainly the richest source of examples of complex systems is found in nature. Biological systems consist of a large number of small interacting components at one level and their interaction gives rise to new structures at a higher level, including e.g. bio-molecules, cells, specific organs, organisms, populations, and finally entire ecosystems. *Morphogenesis,* or formation of structures in nature, is always produced by complex growth processes (Jacob 1994). Biological systems can also be characterized by their adaptive behavior, i.e. their underlying mechanism to adapt and survive in uncertain environments. Hence, they are often referred to as *complex adaptive systems* that can appropriately adapt to the environmental changes. From an engineering point of view, it is important to ensure that engineering designs can adapt to changing environmental conditions because that guarantees their robustness, a required property of almost all engineering products.

One of the most prominent examples of a complex system is human brain. It is arguably the most complex system on the Earth, regarded by many as the 'cathedral of complexity' (Schum 2001). It consists of the order of 10^{10} neurons, and each neuron is connected to 10^3-10^4 other neurons. This extremely complicated spatial topology of neurons and richness of their interconnections can produce incredibly complex behavior that cannot be found nor derived from any set of neurons in the brain.

After almost 40 years of intensive scientific research, the techniques of nonlinear dynamics have been relatively well developed. Nevertheless, a current state of knowledge in this field is largely limited to low-dimensional systems in which there are only a few important dynamical variables. At present, scientific efforts are focused on much bigger, high-dimensional, dynamical systems. Major areas of research include spatio-temporal chaos, synchronization, quantum chaos, pattern formation and complex growth, and time-series analysis. It's impossible to present recent developments in all of these fields even in a very general outline. The following section presents only a short summary of recent developments in this field which are relevant to structural engineering.

2.3.3. Dynamical Systems in Structural Engineering

Engineers are often confronted with nonlinear phenomena and dynamical systems simply because they are confronted with nature. Little has been done so far, however, in terms of research progress in the context of dynamical systems approach to nonlinear engineering problems. Due to complexity of the problems and available computational resources, usually simplified linear models were assumed rather than nonlinear ones. This situation, however, is gradually changing (Thompson 1999). A short summary of several recent applications of dynamical systems and chaos theory in structural engineering is presented below.

Elastic buckling is a nonlinear problem of great importance for engineers. Almost all engineering structures that are designed and built nowadays must be checked against buckling resistance. This problem is has been recently analyzed from a dynamical systems perspective by Holmes et al. (2000; 1999). In their study, the problem of elastic buckling of an inextensible rod with free ends has been investigated. The rod was confined to the plane and subjected to distributed body forces derived from a potential field. This boundary value problem (BVP) may be written as a Hamiltonian system with three degrees of freedom. Numerical studies performed by Holmes et al. revealed that this system has chaotic solutions. They have investigated local bifurcations of these solutions as well as homoclinic and heteroclinic orbits.

In the example described above, dynamical systems and chaos theory have been successfully applied to a boundary value problem in nonlinear mechanics. This new approach brings a broader/holistic understanding of mechanical phenomena in a sense that it generates a "global picture" of a behavior of an engineering system.

Another interesting research problem, which is related to structural engineering, involves excited heated plate systems. Such systems are used for example in hypersonic skin panels in transatmospheric vehicles and have been thoroughly studied by many scientific teams. Recently, Fermen-Coker et al. (2000) conducted numerical simulations using dynamical systems approach to analyze chaotic oscillations that occur under various external loading and boundary conditions. They have also examined the impact of the panels' geometry on the system's dynamic response. Chaos has been detected through the computation of Lyapunov exponents (Alligood et al. 1996). They have also considered other parameters which affect the panels' dynamical behavior: their size and aspect ratio as well as their thickness. It has been found that the critical temperature increase varies significantly when the aspect ratio is close to 1, i.e. when the panels are square. The closer the panel to the square shape, the more sensitive the critical buckling temperatures are to the changes in the aspect ratio of the panel. This effect is increased if the plate dimensions are smaller.

Research in aerospace engineering on controlling or prevention of chaotic oscillations in real-world systems has been mainly focused on the use of adaptive structural concepts (Hall II and Hanagud 1991). Fermen-Coker et al. study brings a new understanding and a global picture of this design space. It is aimed to establish new design rules that can be incorporated at the design stage, so that the use of active control may be minimized, or if possible, eliminated. Their research resulted in discovering interesting relations between the panels' geometry, the critical temperature, and the amplitude of the excitation force that can be used in a design process. It is a good example of how this new approach can be successfully applied to engineering and may bring new tools for engineers.

2.3.4. Dynamical Systems Model of a Simple Genetic Algorithm

As I discussed it earlier, evolutionary algorithms are also instances of complex adaptive systems. As such, they can be modeled and analyzed using available dynamical systems theory and tools. One of such models considers a simple version of a genetic algorithm, called a simple genetic algorithm (SGA), and is due to Vose (1999b). The model establishes a mathematical framework in which actions of basic genetic operators like proportional selection, mutation, and crossover can be analyzed. In its initial formulation (Vose 1990), the dynamical system model of an SGA considered only binary representations, mutation determined by rate, one-point crossover, and used a simplifying assumption of infinite population. Subsequently, the model has been extended and generalized to Random Heuristic Search, which is sufficiently general to describe a variety of search methods including genetic algorithms, simulated annealing, and genetic programming (Vose 1999a).

Vose's model of an SGA considers a population of solutions as a point in the *space of population vectors*. Then the action of the SGA moves this population into another point in this space. The points visited by subsequent generations of the SGA form a trajectory. Hence, we can consider the action of the SGA as a discrete dynamical system.

The action of the SGA from generation to generation is determined by different genetic operators. The action of the proportional selection, mutation, and crossover are modeled as mathematical operators (matrices) which act upon a population vector and move it to another point in the population space. Thus, for example the action of the proportional selection is defined as a diagonal matrix, where each entry defines a ratio of the fitness of each individual and the average fitness of all individuals in the population.

There are three principle conjectures concerning the dynamical model of the SGA (Rowe 2001):

- "It is focused under reasonable assumptions about crossover and mutation, that is, given any population vector the sequence converges to a fixed point. This is known to be true if the mutation is defined bitwise with mutation rate < 0.5 and there is no crossover (Vose 1997).
- Fixed points are hyperbolic, meaning that the differential at the fixed point has no eigenvalue with modulus equal to 1.
- Any operator on the population space is well-behaved, meaning that it always maps volumes into other volumes (image of a volume never has fewer dimensions than that volume)."

Assuming that an SGA is focused, well-behaved, and has hyperbolic fixed-points, the following properties hold (Rowe 2001):

1. "There are only finitely many fixed-points.
2. The probability of picking a population vector such that iterates of SGA applied to this vector converge to an unstable fixed-point is zero.
3. The infinite population SGA converges to a fixed-point in logarithmic time."

Thus, the majority of infinite population models of the SGA always seem to converge to a fixed point. However, as Wright and Bidwell (1997) have shown, the SGA can also exhibit a stable cyclic behavior corresponding to untypical mutation and crossover distributions. In another paper, Wright and Agapie (2001) discovered that the infinite population model can exhibit a stable cyclic/chaotic behavior, which implies that the heuristic is not focused. They also suggest that the chaotic behavior can be useful for restoring diversity in a run of a genetic algorithm that is not making any progress.

2.4. Overview of Engineering Design

2.4.1. Engineering Design

Engineering design and design in general have a long history which can be traced back to the beginnings of the humankind. A brief history of design and its distinction from a craft can be found in (Cross 1989). SOTA reviews of engineering design, design theories, methodologies, and methods are discussed in (Antonsson and Cagan 2001; Cross 1984; Dym 1994; Dym and Little 2000; Finger and Dixon 1989a; Finger and Dixon 1989b; Gero 2002; Newsome et al. 1988; Pahl and Beitz 1996; Tong and Sriram 1992a; Tong and Sriram 1992b; Tong and Sriram 1992c).

There are many definitions of engineering design that have been suggested by various researchers (Dixon 1987; Dym 1994; Mostow 1985; Simon 1981). A definition that is most closely related to the proposed research was proposed by Dym (1994):

"*Engineering design* is the systematic, intelligent generation and evaluation of specifications for artifacts whose form and function achieve stated objectives and satisfy specified constraints."

Dym (1994) describes engineering design problems as open-ended and ill-structured. Design problems are said to be *open-ended* because they usually have many acceptable solutions, and hence the uniqueness assumption required by many traditional analysis techniques cannot be satisfied. *Ill structure* of design problems is caused by the fact that their solutions cannot be normally found by routinely applying a mathematical formula in a structured way.

The following two sections present recent developments in engineering design methods and conceptual design.

2.4.2. Engineering Design Methods

Research on design theories, methodologies, and methods is one of the most active fields of research in various subfields of engineering. Good introductions to the field can be found in (Dym and Little 2000; Finger and Dixon 1989a; Finger and Dixon 1989b; Waldron and Waldron 1991). Two main research areas in this field include *methods of representing artifacts* (engineering systems) as well as *methods of modeling design processes*, i.e. processes by which designs are completed. I discussed SOTA in method of representing engineering systems in the context of evolutionary design in section 2.1.3. Recent developments in methods of modeling engineering design processes are presented below.

The proposed models of *engineering design processes* can be generally divided into three categories:

- Formal models
- Heuristic models
- Agent-based models

The goal of *formal approaches* to modeling design processes is to establish engineering design science (Dixon 1987; Suh 1990). A systematic approach to engineering design has been first proposed by Pahl and Beitz (1984). In their seminal work, four major phases of engineering design are distinguished: clarification of the task, conceptual design, embodiment design, and detailed design. Other models of major phases of a design process were proposed by French (1992) and Cross (1989). An axiomatic approach to engineering design process was introduced by Suh (1990). He proposed axiomatic design theory (ADT) which is based on two axioms: independence axiom and information axiom. Formal grammars, both context-free and context-sensitive, have been introduced to engineering design by Mullins and Rinderle (1991; Rinderle 1991). Shape grammars have been suggested by Stiny (1980) and applied to mechanical design problems by Schmidt and Cagan (1994). Finally, TRIZ (a Russian acronym for the theory of inventive problem solving discussed earlier in section 2.1.2) proposed by Altshuller (1969; 1999) provides a systematic methodology of creative problem solving in engineering design.

Heuristic models of an engineering design process are based on the generate-and-test, or trial-and-error, method. Here, several methods have been proposed. Arciszewski (1977) proposed morphological analysis in conceptual design. Protocol analysis based on empirical data was used by Stauffer et al. (1987) in mechanical design. Simulated annealing with shape grammar representations was introduced by Shea et al. (1997). Roston (1994) proposed genetic design, a design method based on formal grammars and genetic algorithms/genetic programming. Evolutionary computation has a long history in modeling design processes which was introduced earlier in section 2.1.2. Cellular automata representations combined with evolutionary algorithms have been proposed in the context of architectural design by Frazer (1995). Chan et al. (2000; 2002) have recently introduced an evolutionary framework with dynamic hierarchical representations to enhance design process.

Agent-based models of an engineering design process provide modular, distributed, and knowledge-based approach to solving design problems. Multi-agent design systems (MADS) can address some important issues in a design process, including constant evolution of standards and technologies, dynamic marketplace demands, and high degree of adaptability. On the other hand, additional challenges have to be properly resolved, including interoperability among heterogeneous agents, coordination of design processes, and managing conflicts (Lander 1997). Research in MADS can be traced back to the mid 1990's. Shen and Barthes (1995) proposed a multi-agent environment for engineering design. D'Ambrosio and Birmingham (1995) used preferences and agents in conceptual design. Parmee (1996) introduced a multi-agent system for design space decomposition. Campbell et al. (1999) proposed agent-based approach to conceptual design in dynamic environments. Cvetkovic (2000) used agents for a multiobjective decision support system in conceptual design. Gero and Kannengiesser (2003) proposed a function-behavior-structure framework for situated agents. Current SOTA reviews can be found in (Gero and Brazier 2002; Lander 1997; Shen and Norrie 1999).

2.4.3. Conceptual Design

Conceptual design forms an initial stage of a design process. Pahl and Beitz (1996) define the conceptual design in the following way:

"*Conceptual design is that part of the design process in which, by the identification of the essential problems through abstraction, by the establishment of function structures and by the search for appropriate working principles and their combination, the basic solution path is laid down through the elaboration of a solution principle. Conceptual design determines the principle of a solution.*"

Dym (1994) defines conceptual design from a more computational perspective. He considers the conceptual design phase of a design process as a phase that "… has as its output a concept. This goal is achieved by:

- identifying the most crucial or essential problems
- establishing a function structure, i.e. a framework in which the artifact will perform its primary function, including a decomposition of the primary function into subfunctions that will be performed by subsystems or individual components
- formulation of a solution procedure that can be successfully applied to the design problem
- preparing concepts or skeleton designs or schemes
- evaluating candidate schemes against the relevant criteria, including both economic and technical matters."

Recently Arciszewski et al. (2003) proposed a new approach to conceptual design in the context of chaos. They distinguish two types of conceptual design processes based on the number of design concepts generated. The first type (Type I) occurs when only a limited number (one or a few) of design concepts are produced. This type corresponds to a traditional approach to conceptual design. The second type (Type II) appears when many design concepts are generated, on the order of thousands or even hundreds of thousands. The latter type is equivalent to a situation when a designer uses various conceptual design support tools.

When conceptual design process is analyzed as a process of learning, or acquiring engineering knowledge, then a Type I process results in a limited amount of additional background knowledge. Therefore, this type can be called a *point design*, because only a limited number of points in the design space can be identified, where each point corresponds to a single design concept. On the other hand, a Type II process allows one to acquire significant amount of additional background knowledge. Thus, it can be called a*picture building design*. This picture is created considering together a large number of points in the design space. All these points taken together form an invaluable picture of a given engineering design domain and give a new insight into it on a conceptual level.

The Type I conceptual design process has been the subject of design research for the last 30 years that resulted in many design theories and methods (see section 2.4.2). However, very little is known about the Type II process, although, as Arciszewski et al. suggest, it will become soon a predominant type of a conceptual design process.

They also propose a chaos-based model for the Type II conceptual design process. It has three major assumptions:

- "It is a search process, conducted through the design space, for design concepts that satisfy given requirements and constraints.
- It can be classified at the Type II conceptual design process, which involves the generation of a large number of design concepts and acquiring background knowledge.
- It can be analyzed using concepts and models developed in the field of chaos."

In their view, a conceptual design process can be considered as a *dynamical search process* iterated many times, and each time a design concept is produced (corresponding to a single point in the design space). The generation of the individual design concepts is driven by a specific mechanism, which can be a heuristic method or an evolutionary computation algorithm. Such a single concept generation can be assumed as occurring on the local level of the conceptual design process. When all generated points in the design space are analyzed together, then the global picture of the conceptual design process emerges. This picture can be assumed as appearing on the global level of the conceptual design process, and it can be studied from the dynamical systems point of view.

When one uses a dynamical systems approach to analyze a design process, then the well-known design concepts can be treated as attractors in the design (search) space. It has been proved (Clarke 2000) that such well-known design concepts attract the attention of human designers. This phenomenon has been described as an *inertia vector*. Chaos-based model of conceptual design implemented in a design support tool could be then focused on minimizing the influence of inertia vector (avoiding the basin of attraction of a well-known design concept) and searching the design space for novel designs.

2.4.4. Design of Structural Systems in Tall Buildings

The empirical validation of the design method proposed in this dissertation (see section 3.6 for a detailed description of the entire validation process) was conducted using two classes of design problems in the domain of steel structural systems in tall buildings. Hence, in this section I introduce the problem of designing steel structural systems in tall buildings and discuss its major characteristics.

Steel skeleton structures in tall buildings are considered the most complicated structures designed and built. Their conceptual and physical complexity can only be compared to such complex structural systems as, for example, large span bridges or large span space structures. Usually, steel structural systems in tall buildings are designed as a system of vertical members called *columns*, horizontal members called *beams*, and various diagonal members called *wind bracings*, since they are added to columns and beams to increase the flexural rigidity of the entire system and that is driven mostly by stiffness requirements related to wind forces.

Skeleton structures are designed to provide a structural support for tall buildings. They have to satisfy numerous requirements regarding the building's stability, transfer of loads, including gravity, wind and earthquake loads, deformations, vibrations, etc. For this reason, the design of structural systems in tall buildings requires the analysis of their behavior under various combinations of loading and the determination of an optimal configuration of structural members. It is difficult, complex, and still not fully understood domain of structural engineering.

The two design problems considered in this dissertation represent two classes of design problems characterized by distinct levels of their structural complexity. First, a relatively simpler problem of designing a wind bracing system in a tall building is investigated. In this problem, an optimal configuration (topology) of wind bracing members in a steel structure is sought assuming the same *configurations* of beams, columns, and supports throughout an entire design process. In this case, however, cross-sections of *all* members (including beams, columns, and wind bracings) are optimized during the detailed design stage (sizing optimization). Next, this problem was extended, and novel as well as optimal configurations of the entire structural system, which includes beams, columns, wind bracings and supports, were sought.

The two design problems described above exhibit some important properties shared by a much larger class of problems in structural engineering and engineering design in general. First, both classes of investigated engineering systems consist of a relatively large number of *identical*, *simple*, and *locally interacting* structural members. For example, in the problem of designing an optimal wind bracing system, a configuration of wind bracing members is represented by attributes which assume **identical** sets of possible values. Also, the configuration of wind bracing members forms a **simple** and **uniform** structural grid contained within adjacent vertical and horizontal grid lines defined by columns members and beam members, respectively. The wind bracing members **locally interact** with each other and with the column and beam members placed in their neighborhood. A global behavior of the structural system emerges from the local interactions of all structural members from which the steel structure is formed.

In the conducted design experiments, the process of design is considered as a three-stage process. The first stage, design concept generation, produces a design concept, or a class of design concepts. By this term, an abstract description of a future structural system is understood, and it is provided in terms of qualitative, or symbolic, attributes. The second stage, topology/shape optimization, identifies an optimal configuration (topology) of a structural system, the nature of connections, materials, etc. Finally, the third stage (sizing optimization), produces a detailed design, and it involves the structural analysis, dimensioning, and numerical optimization. A detailed discussion on the phases of the design process assumed in this dissertation is presented in section 3.4.

One of the most difficult and important parts of the design process is the determination of an appropriate configuration (topology) of a structural system for a given building. In terms of novelty and weight of a structural system, the optimization of its configuration is much more important than the final numerical optimization of individual structural members, or even of an entire structural system, if an incorrect design concept is selected. Traditionally, however, due to the complexity of this problem, a structural system configuration is selected considering only very few design concepts which are not necessarily optimal for a given building (Mustafa and Arciszewski 1992). In this dissertation, I address this problem by proposing novel mechanisms of generation of design concepts and their optimization based on models of complex systems (see chapter 3).

2.5. Overview of Epistemology and Validation of Design Methods

2.5.1. Historical and Philosophical Perspective

Epistemology (the theory of knowledge) has a long history that can be traced back to Phyrro and his skeptics in ancient Greece. Historically, two major philosophical schools emerged regarding the criterion for truth and the validation of new scientific knowledge:

Foundationalist/Formalist/Reductionist School of Epistemology and Holistic/Social/Relativist School of Epistemology (Barlas and Carpenter 1990).

Foundationalist school started by Aristotle and Plato assumes that there are absolute truths that are independent of time, place, and context. Two major trends emerged within this school, including *rationalism* and *empiricism*. Rationalism was founded by Descartes ([1641] 1931) who asserted that "the truth is innate and prior to all experience" and "human knowledge about the truth is based on reasoning." The latter assertion was challenged by Locke ([1690] 1894), the father of the empiricism, who argued that "all human knowledge about the truth is based on experience rather than reasoning." More recently, foundationalist view was reintroduced by Russell (Russel 1962; Slater 1988) and his *logical atomism*, and Wittgenstein ([1921] 1961), the father of *logical positivism*. Logical positivism asserted that "knowledge can only be claimed if judged true by meaning [analytically true] or true by virtue of experience [synthetically true]" (Pedersen et al. 2000).

Foundationalist views were first questioned by Kant ([1781] 1933) who asserted that "all knowledge starts from experience" but "not all knowledge arises out of experience." Kant first suggested that not all truths are innate and absolute. His views were later extended by Hegel ([1817] 1959) who totally rejected the idea of innate truths and introduced a new logic, called the *coherence theory*. In Hegel's logic conflict and contradictions are regarded as necessary elements of truth. He regarded *truth as a process* and not as a fixed state of things. In his view *knowledge is socially, culturally, and historically dependent* and hence entirely objective verification of knowledge claims in not possible. More recently, foundationalist view was challenged by Kuhn ([1962] 1970) who presented a historical analysis of how science progresses and argued that a change to a new paradigm in science cannot be based strictly on logical reason. The second part of the last century brought further attacks on empiricism and led to the *relativist* philosophies of science (Barlas and Carpenter 1990).

The two schools of epistemology discussed earlier formed two opposing philosophies of validating new scientific knowledge. First, logical empiricist validation, founded by the Foundationalist/Formalist/Reductionist School of Epistemology, is a "strictly formal, algorithmic, reductionist, and 'confrontational' process, where new knowledge is either true or false. The validation becomes a matter of formal accuracy rather than practical use. This approach is appropriate for closed problems that have right or wrong answers associated with them, like mathematical expressions or algorithms" (Pedersen et al. 2000). Relativist validation, on the other hand, is based on the Holistic/Social/Relativist School of Epistemology, and can be defined as a "semiformal and communicative process, where validation is seen as a gradual process of building confidence in the usefulness of the new knowledge (with respect to a purpose). This approach is appropriate for open problems, where new knowledge is associated with heuristics and non-precise representations" (Pedersen et al. 2000).

As I discussed it in section 2.4.2, engineering design problems are open-ended (have many acceptable solutions) and ill-structured (solutions cannot be normally found by routinely applying a mathematical formula in a structured way). Thus, relativist approach is more appropriate for validation of design methods (Pedersen et al. 2000) and hence it was selected to validate Emergent Engineering Design, the design method proposed in this dissertation. The follwoing section introduces a framework for validating design methods, called the Validation Square. A detailed description of the validation methodology (based on the Validation Square framework) which I use in this dissertation to validate the proposed design method is presented in section 3.6.

2.5.2. Validation Square – A Framework for Validation of Design Methods

Validation of traditional analytical engineering research, based on mathematical modeling, has been mostly conducted using a logical empiricist validation approach. Design methods for engineering design, however, rely not only on mathematical modeling but also on subjective statements and various heuristics. Engineering design is mainly concerned with open problems that involve both objective and subjective elements and have no single right answer (non-uniqueness). It requires both science and art to achieve its goal (Pedersen et al. 2000). Hence, relativist validation approach seems to be better suited for validation of design methods.

Hence, research validation in this dissertation is conducted using the relativist approach, and more specifically using the Validation Square methodology recently proposed to validate design methods and research (Pedersen et al. 2000). The validation strategy assumed in this dissertation is based on the following statement:

"Scientific knowledge in the field of engineering design is defined as socially justifiable belief according to the Relative School of Epistemology. It is due to the open nature of design method synthesis, where new knowledge is associated with heuristics and non-precise representations. Thus, knowledge validation becomes a process of ***building confidence in its usefulness*** *with respect to a purpose"* (Pedersen et al. 2000).

The process of validation of a design method, according to the Validation Square framework, is shown in Figure 12 (adapted from (Pedersen et al. 2000)). As stated earlier, the validation of new scientific knowledge is a process of building confidence in the usefulness of the proposed design method with respect to a purpose. The usefulness of a design method is associated with the two major criteria:

- Effectiveness - the method provides design solutions correctly.
- Efficiency - the method provides correct design solutions.

Correct solutions are understood in this context as solutions with acceptable performance. Effectiveness provides a qualitative evaluation of the design method while efficiency gives its quantitative assessment.

Effectiveness of a design method can be realized by conducting the *qualitative* process of *structural validation*. This process consists of three major stages (Pedersen et al. 2000):

- Accepting the individual constructs constituting the method (Theoretical Structural Validity)
- Accepting the internal consistency of the way the constructs are put together in the method (Theoretical Structural Validity)
- Accepting the appropriateness of the example problems that will be used to verify the performance of the method (Empirical Structural Validity)

Efficiency of a design method can be realized by conducting the quantitative process of *performance validation*. This process also consists of three major stages (Pedersen et al. 2000):

- Accepting that the outcome of the method is useful with respect to the initial purpose for some chosen example problem(s) (Empirical Performance Validity)
- Accepting that the achieved usefulness is linked to applying the method (Empirical Performance Validity)
- Accepting that the usefulness of the method is beyond the case studies (Theoretical Performance Validity)

A detailed description of how the Validation Square methodology was used in this dissertation to validate Emergent Engineering Design is presented in section 3.6.

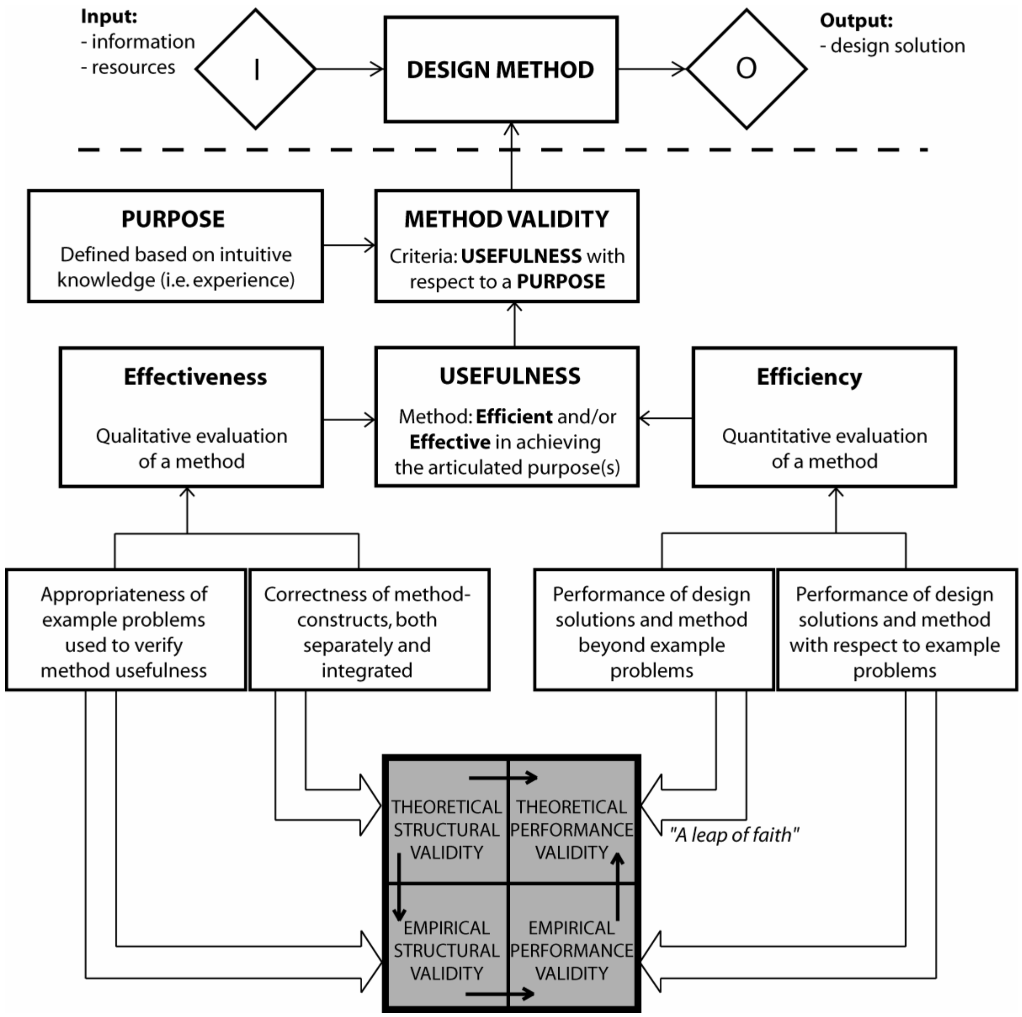


Figure 12. Validation Square framework for validation of design methods

2.6. Summary

In this chapter, I presented relevant background knowledge to provide some context necessary for understanding the rest of this dissertation. The first part of this chapter introduced current research developments in the fields corresponding to the components of the proposed design method, i.e. evolutionary computation, cellular automata, and complex systems. The second part provided an overview of engineering design in general as well as introduced the classes of design problems considered in this dissertation. Finally, the last part of this chapter gave a historical perspective on the problem of validation of new scientific knowledge and described the validation methodology which was used in this dissertation to validate Emergent Engineering Design.

The first section of this chapter introduced evolutionary computation and canonical evolutionary algorithms. It also provided description of the current research developments in the subfields of evolutionary computation related to engineering design. The topics discussed in this section included evolutionary design and creativity, new ways of representing design, methods of handling constraints, multiobjective evolutionary design, and coevolutionary design. Also, together with a comprehensive review of the applications of evolutionary computation in structural engineering, open issues and most promising research directions were discussed.

The second section of this chapter introduced cellular automata and the New Kind of Science. It showed that even very simple programs can produce complex behavior. Four classes of the dynamical behavior exhibited by elementary cellular automata were discussed: fixed-point behavior, periodic behavior, chaotic behavior, and localized propagating structures. Also, a numbering scheme of cellular automata rules was introduced as it will be used extensively in the remaining part of this dissertation. The last part of this section reported several examples of applications of cellular automata in structural engineering.

The third section provided a high-level material on dynamical systems, chaos theory, and complex adaptive systems. Also, a few applications of dynamical systems and chaos theory in structural engineering were presented. The last part of this section introduced a dynamical systems model of a simple genetic algorithm to show applicability of these theoretical tools to model complex adaptive systems.

The fourth section of this chapter introduced engineering design and presented a classification of existing design methods. It also presented in more detail an initial stage of the design process, called conceptual design, which will be specifically addressed in this dissertation. Finally, two conceptual design problems in structural engineering, namely conceptual design of wind bracing systems and conceptual design of the entire steel structural systems in tall buildings, we described. They will be used in the remainder of this dissertation to empirically validate Emergent Engineering Design.

The last section of this chapter provided philosophical and historical perspective on validation of new scientific knowledge. Also, the Validation Square, a framework for validation of design methods, was introduced and described in detail. This framework will be used in this dissertation to validate Emergent Engineering Design.

Having provided relevant background knowledge, I can now introduce Emergent Engineering Design, the design method proposed in this dissertation. It will be described in detail in the next chapter.

3. EMERGENT ENGINEERING DESIGN

"Living organisms are examples of design strictly for function, the product of blind evolutionary forces rather than conscious thought, yet far excelling the products of engineering. When a designer looks at nature he sees familiar principles of design being followed, often in surprising and elegant ways. Sometimes, as in the case of flight, he is inspired to invention: more commonly, he discovers his ideas embodied in some animal or plant."

(Michael French)

In this chapter, I propose and define Emergent Engineering Design, a design method based on models of complex systems and the main objective of this dissertation. In the first part of this chapter, in sections 3.1 and 3.2, I state the problem addressed by the proposed design method and relate it to the open issues in the field of engineering design which I identified in the background review presented in chapter 2. Next, I define Emergent Engineering Design and introduce the structure of the argument presented in this dissertation in the form of research questions and the corresponding research hypotheses. I start section 3.3 with the fundamental question and the fundamental hypothesis of this dissertation and subsequently decompose them into research questions and research hypotheses corresponding to the major phases of a conceptual design process.

Furthermore, in section 3.4, I discuss the scope of research reported in this dissertation and provide a detailed description of the assumptions incorporated in the proposed design method. I also provide an outline the conducted research in section 3.5. In the last part of this chapter, in section 3.6, a detailed description of the validation methodology which was used to validate Emergent Engineering Design is presented. Figure 13 shows an organization chart of chapter 3 with all sections discussed above.

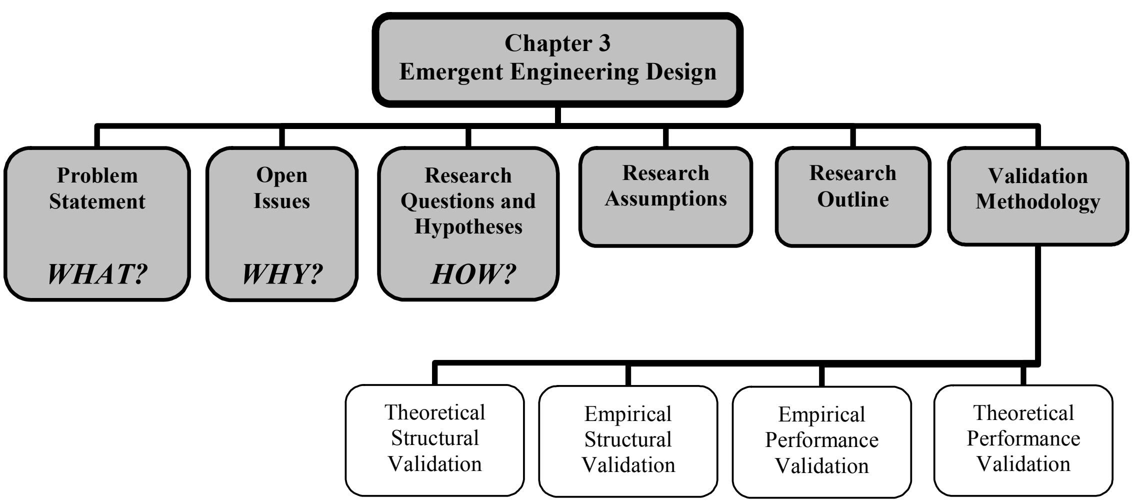


Figure 13. Organization of chapter 3

3.1. Problem Statement

The underlying problem I addressed in this dissertation can be stated in the following way: how to establish a method for designing engineering systems based on models of complex systems and inspired by the processes occurring in nature? If such a method can be discovered, developed, and implemented, it may be applied to a broad spectrum of engineering design problems. It may also provide a new understanding of engineering design as well as significantly enhance traditional engineering design processes and achieve their two important objectives: development of novel designs and their optimization.

Development of a new design method implies a general framework for doing design of engineering systems. Thus, it cannot be limited to a particular sub-domain but it must be applicable to a broad spectrum of engineering problems. Generality of the proposed method should also encourage its use at all stages of a design process, i.e. it must be applicable to conceptual design, embodiment design, as well as detailed design (see section 2.1.2). The only modifications required at each stage would involve appropriate tuning of the representation accuracy (granularity) and its nature (symbolic, symbolic + numerical, numerical), concept generation mechanisms (more generative and creative versus more parameterized and optimization oriented), and an evaluation procedure.

At first, the problem statement and scope of the research seem to be too ambitious and difficult to achieve. It is my opinion, however, that all the required pieces necessary to succeed are already available and they have to be properly assembled and integrated. It is the appropriate synthesis/fusion of knowledge coming from computer science (algorithms, data structures), engineering (design representation spaces, design evaluation), mathematics (dynamical systems theory, chaos theory, models of various complex systems), and biology (processes of natural selection, evolution and coevolution) that made this goal achievable.

The proposed design method requires some domain specific knowledge in a form of a design representation as well as a design evaluation procedure. However, the process of building a representation of an engineering system should not be as tedious as it is the case with some previous approaches, e.g. formal grammars (Roston 1994), or shape grammars (Schmidt and Cagan 1998). Also, the issue of providing a proper design evaluation procedure can be handled quite efficiently. In structural engineering, for example, a single structural analysis package would be able to evaluate many design concepts, including simple frames and trusses as well as complex steel structural systems in tall buildings.

It is, of course, impossible to investigate the proposed design method in its entirety within the timeframe of the dissertation. Hence, several decisions were made to restrict the scope of research and to make it more manageable. Thus, in this dissertation I investigate Emergent Engineering Design in the context of conceptual design problems only (see section 2.4.3). Furthermore, the applications and empirical validation of the proposed design method are restricted to structural engineering problems (see section 2.4.4).

3.2. Open Issues

As I discussed it in sections 2.1.3 and 2.4.2, there have been many suggested approaches to develop methods for engineering design. However, in my opinion, most of them were focused exclusively on only one of the two important aspects of engineering design, i.e. either on creativity or on optimization. Hence, there is a need to develop a method that could account for both of these aspects.

Another issue is that many of the proposed design methods tended to be assembled from conceptually diverse components and thus not giving a coherent view of a design process. It is my belief that it is important, and at the same time possible, to develop a coherent engineering design method based on models of complex systems.

Yet another relevant, and so far unexplored, issue is the possibility of modeling natural phenomena using simple programs rather than systems of complicated partial differential equations, as it has been done in traditional science. As suggested by Wolfram (2002), complex systems modeled by simple programs might provide completely new understanding of many processes and phenomena.

One of the major properties of complex systems is the richness of local interactions among the systems' elements and their emergent behavior. This issue has been investigated by several researchers in the context of engineering design, mainly in architectural design (Chan et al. 2002; Frazer 1995; Poon and Maher 1996b). There are, however, very few applications of these ideas in civil and structural engineering. On the other hand, structural engineering systems are known to exhibit large sensitivity to local interactions among structural elements. Thus, further investigation of the potential of representing structural systems using simple rules that model local interactions is highly justifiable. It may provide a more qualitative and holistic approach to this traditionally strictly quantitative and optimization-oriented field.

3.3. Research Questions and Hypotheses

The argument for this dissertation is structured in a way that corresponds to the Scientific Method. The structure of this argument has been adapted from (Pedersen 1999). Research questions are used to determine research issues. Research hypotheses provide an intellectual value to the research supporting answers to research questions. Hypothesis testing is employed to justify claims of contribution to the field in which the research is conducted (Pedersen 1999). When presented in the context of Natural Science, research questions correspond to observations (articulating the 'truth'), hypotheses correspond to explaining the observations (understanding the 'truth'), and hypothesis testing corresponds to validating the explanation (accepting knowledge about the 'truth') (Pedersen 1999). Hence, as it is argued in Pedersen (1999), "in the context of engineering design, hypothesis-testing becomes the vehicle by which new scientific knowledge is accepted and added to the current pool of knowledge. This ties research validity discussion strongly to a fundamental problem addressed early in epistemology and later in the philosophy of science: *what is scientific knowledge, and what constitutes confirmation of a knowledge claim?*"

The formulation of the fundamental research question this dissertation attempted to answer was motivated by the problem stated in section 3.1, the nature of the problem outlined in chapters 1 and 2, as well as open issues in the field of engineering design discussed in section 3.2. The fundamental research question can be expressed in the following way:

Fundamental Research Question

How can one construct an effective method for designing engineering systems that would support development of novel designs and their efficient optimization?

Correspondingly, the formulation of the fundamental research hypothesis of this dissertation is based on the concepts of complex systems discussed in section 2.3.2 and inspired by the processes and phenomena occurring in nature. The fundamental hypothesis is formulated in the following way:

Fundamental Research Hypothesis

Emergent Engineering Design, a design method in which all major elements of engineering design (i.e. design representation, actual design process, and design evaluation) are modeled as complex systems, can effectively produce novel designs and efficiently optimize them.

Given the fundamental research question and the fundamental research hypothesis, the ultimate objective of this dissertation can be expressed in the following way:

Ultimate Dissertation Objective

Develop an engineering design method based on models of complex systems that provides a conceptually coherent framework for producing novel designs and their efficient optimization.

The proposed design method has been named Emergent Engineering Design (EED). The method is understood here as a basic conceptual system consisting of a class of models, procedures, and algorithms for engineering design. The validity of the proposed method will provide an answer to the fundamental question.

The ultimate dissertation objective is, out of necessity, very general. In order to facilitate the development of the proposed method in a more structured way, it has been divided into four sub-issues pertaining to:

- Identification of mechanisms to accomplish design novelty (see discussion on creativity and design in section 2.1.2)
- Determination of effective ways of decomposing complex design problems into sub-problems
- Selection of efficient optimization mechanisms
- Establishing mechanisms to evaluate designs in a way to guarantee their robustness

The four issues discussed earlier are addressed in the following research questions numbered 1 to 4, respectively.

Research Question 1 (Represent):

Based on the existing knowledge on how to represent engineering systems; what mechanisms and models can be used to produce novel designs?

Research Question 2 (Decompose):

Knowing that complex engineering design problems are usually decomposed into sub-problems; how can a decomposition of an engineering system be defined and how can a decomposed system be effectively designed?

Research Question 3 (Generate and Optimize):

One of the major objectives of almost all engineering design processes is achieving optimality; what mechanisms should be used to efficiently optimize engineering designs?

Research Question 4 (Evaluate):

Evaluation of design concepts is one of the most important stages of a design process; how can the evaluation process be performed to accomplish robustness of designs?

Finding answers to these questions is coupled with a successful development of the proposed method of engineering design. The models, procedures, and algorithms discussed in chapters 1 and 2 form hypotheses upon which the EED is built. The formulation of the four research hypotheses, corresponding to the previously stated research questions, is presented below.

Research Hypothesis 1 (Represent):

Evolutionary design and complex systems provide a framework for defining generative representations, i.e. representations of engineering systems based on simple programs, which can successfully produce novel designs.

Research Hypothesis 2 (Decompose):

Cooperative coevolutionary models provide an efficient framework for a decomposition of complex design problems and conducting design processes using cooperative coevolutionary algorithms.

Research Hypothesis 3 (Generate and Optimize):

Evolutionary computation provides a framework for conducting engineering design processes and optimizing engineering designs.

Research Hypothesis 4 (Evaluate):

Competitive coevolutionary models are suitable for adaptive testing and evaluation of engineering design concepts and can successfully increase robustness of generated designs.

Selection of the research hypotheses in relation to the posed research questions is based on the discussion and justification presented in chapters 1 and 2. In the Scientific Method, research questions are answered when the corresponding research hypotheses are tested and resist being invalidated. In this dissertation, the research questions and corresponding research hypotheses are incorporated in the structure of EED, as it is shown in Figure 14.

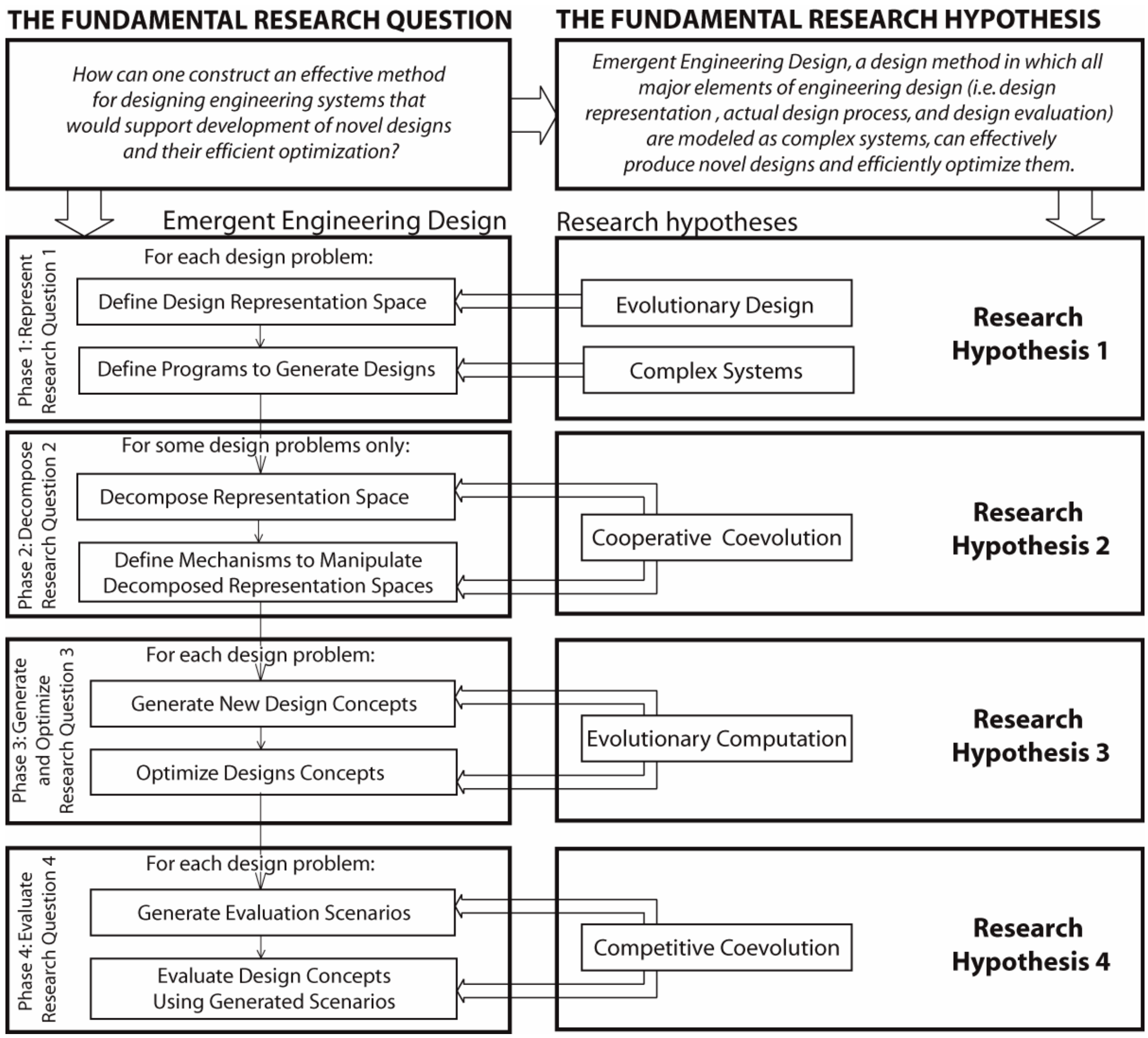


Figure 14. Phases of Emergent Engineering Design and their relationship to four research questions and corresponding research hypotheses

Figure 14 shows that answering the fundamental research question is coupled with the validation of the supporting research hypotheses, which in turn is coupled with the validation of Emergent Engineering Design. A detailed description of the validation methodology will be presented later in section 3.6.

Even though the fundamental research question and the fundamental research hypothesis have been considerably refined, the resulting four research questions and four research hypotheses are still quite general and rather vague. However, at this stage, they need to be formulated in this manner due to the intended generality of the proposed design method. In other to further refine them, some domain and problem specific information must be added, e.g. we have to define what we mean by a novel, or optimal, design, and an effective, or efficient, design process. Chapters 6-8 show the process of refinement of the research questions 1 and 3 for a specific domain (structural engineering) and specific design problems (conceptual design of wind bracing systems and conceptual design of steel structures in tall buildings). The refined research questions will be sufficiently precise to form the basis for hypotheses that can be tested and possibly falsified.

In the remainder of this dissertation, I investigate the core of the proposed design method, i.e. phases 1 (represent) and 3 (generate and optimize). Phases 2 (decompose) and 4 (evaluate) will become a part of the future work.

3.4. Research Assumptions

This section presents research assumptions incorporated in the proposed method regarding the major phases of an engineering design process and a level of generality of design representations.

First set of assumptions is related to the process of modeling of a conceptual design process. EED assumes four major phases of conceptual design in engineering. The phases correspond to similar phases in traditional conceptual design, as it is shown in Figure 15.

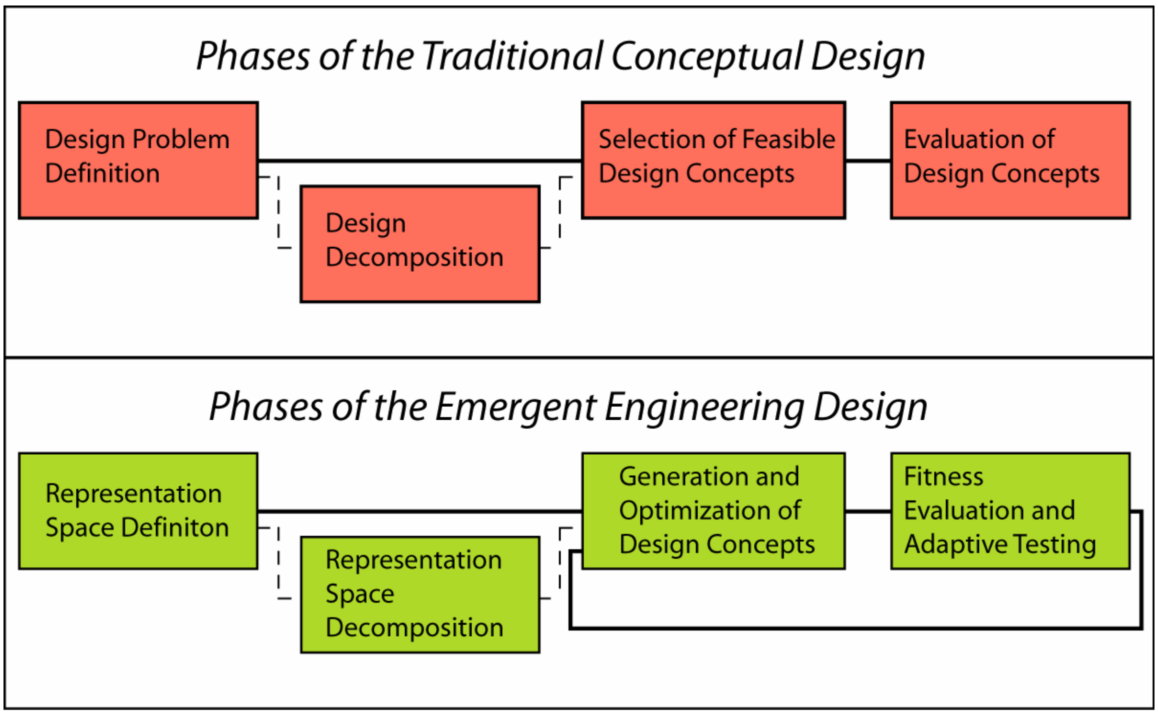


Figure 15. Major phases of Emergent Engineering Design and their relationship to phases of traditional conceptual design

A brief description and comparison of the identified phases in traditional conceptual design and the proposed EED is presented below.

Phase 1

In this phase, an abstract description of an engineering system to be designed is prepared.

Traditional:

Appropriate model of an engineering system is defined, including all requirements and constraints.

EED:

A design representation space is created, and the requirements as well as constraints on the values representing specific attributes are defined.

Phase 2

This phase involves a decomposition of a design problem. This phase is optional and is usually conducted only for complex design problems.

A decomposition of an engineering system can be established at two levels. First, an engineering system's geometry might suggest a possible decomposition of a problem. In this case, the design problem can be divided into a set of decoupled sub-problems which are independently solved and finally assembled together to form a complete design concept. Second, separate functions of various parts of an engineering system might suggest another plausible decomposition. In that case, design components can be identified by performing different functions in an engineering system. For example, in a domain of steel structural systems of industrial buildings designers traditionally decompose the problem into two sub-problems, i.e. design of the main structural system is performed separately from design of a system of wind bracings that assure the rigidity of the structure (usually done after the main structural system is complete).

Traditional:

Design decomposition is performed manually by designers based on geometric or functional criteria using various heuristics and simplifying assumptions.

EED:

Design decomposition is explicitly specified at a representational level by representing each solution to a sub-problem as a separate individual and subsequently coevolving the individuals in several populations.

Frequently, phases 1 and 2 are combined together and treated as one phase of the conceptual design, called problem formulation (Arafat et al. 1992). In the proposed method, however, they are treated separately because different models and algorithms are used to design complex engineering systems that can be decomposed.

Phase 3

In this phase, feasible design concepts, represented by models/representations specified in phases 1 and 2, are generated and optimal solutions are sought.

Traditional:

Selection of feasible concepts is performed. Usually, a human designer considers only a few alternative design concepts.

EED:

Thousands of designs concepts are generated automatically. Here, various complex systems modeled by simple programs, e.g. one-dimensional and two-dimensional cellular automata, are used to generate design concepts (representations of engineering systems) and evolutionary and cooperative coevolutionary algorithms to perform design optimization (representations of design processes).

Phase 4

In this phase, evaluation of design concepts, generated earlier in phase 3, is conducted.

Traditional:

A structural analysis package is used to perform simulation of behavior of an engineering system represented by a design concept. Its subsequent evaluation is performed by the designer.

EED:

The evaluation process is combined with generation and optimization of engineering design concepts. When a new generation of design concepts is produced, the behavior of individual designs is analyzed/simulated by a structural analysis package. The process of analysis/simulation of behavior of engineering systems represented by design concepts requires one, or sometimes more, evaluation scenarios. They can be determined in the following ways:

- Standard evaluation scenarios which are used in practice, e.g. types, locations, and magnitudes of loads and load combinations determined by structural design codes.
- Coadapted evaluation scenarios which are coevolved together with design concepts. In this process, called *adaptive testing*, competitive coevolutionary algorithms are used.

  In the competitive coevolutionary model, two competing populations are coevolved: a population of design concepts (topologies of structural systems) and a population of evaluation scenarios (locations, types, and magnitudes of loads and load combinations). The two populations coevolve in the following way. The fitness of each individual design in the population of design concepts is determined by measuring how well it performs against the evaluation scenarios from the population of scenarios. On the other hand, the fitness of each scenario depends on the number of design concepts it "defeated," i.e. how many designs didn't succeed to satisfy design requirements (like max. stresses, max. horizontal displacement, etc.) under this loading case.

  This approach seems more natural than standard load tables, codes, and loading combinations traditionally used in structural engineering. It might be indispensable when robustness of designs is one of the key issues. Robustness of the design concepts can be improved, e.g. by testing sensitivity of various engineering systems to certain classes of scenarios that would never be applied by human designers.

In EED thousands of designs concepts are generated and subsequently evaluated. In this case, evaluation phase provides feedback to the algorithm generating and optimizing design concepts, as it is shown in Figure 15.

The four phases described earlier can be related to existing engineering design methods described in section 2.4.2. Phases 1 and 2 correspond to formulating representations of artifacts, phase 3 to modeling design processes, and phase 4 to modeling design evaluation processes.

Another set of assumptions considers the generality of representations of engineering systems. When building a representation space, the assumed level of generality is of great importance and may significantly affect the quality of obtained solutions. As mentioned earlier, in this dissertation, conceptual design of structural engineering systems is considered. The spectrum of possible choices of generality of representations in structural engineering is presented in Figure 16.

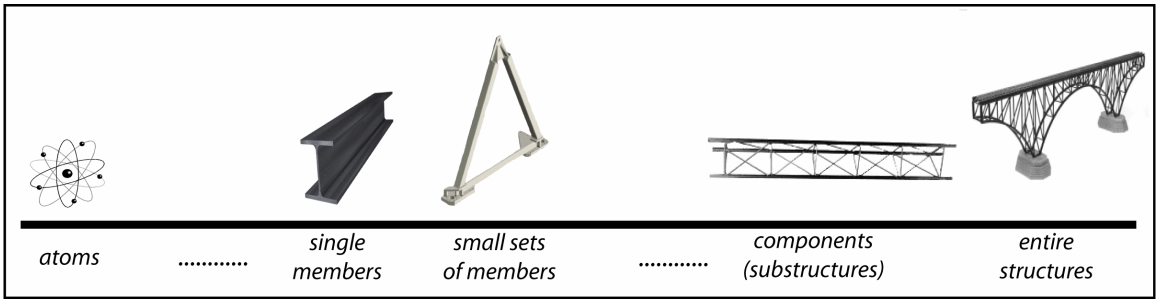


Figure 16. Spectrum of generality of representations of structural systems

The most general representation would consider engineering systems as consisting of individual atoms and a search mechanism would navigate through the space of possible configurations of atoms. This approach is, of course, infeasible with computational power available today. The other end of the representational spectrum consists of entire structures. In this case, the representation simply parameterizes the structure and an optimal set of parameter values is being sought.

One of the major objectives of EED is to accomplish novelty in generation of design concepts. Thus, the level of generality of the representation of an engineering system has to be rather high. It is believed that appropriate results can be achieved when the representation consists of single members, or small sets of members, as basic representational units. On the other hand, emergent structural shaping phenomena are expected to appear at the component, or substructure, level.

3.5. Research Outline

Research described in this dissertation has been divided into four parts, which partially overlap with the major phases of Emergent Engineering Design described in the previous section.

- **Part One – Selection of an Engineering Domain and Building Representations**

  Part one was devoted to choosing a structural engineering domain and building appropriate representations of engineering systems. The issues involved at this stage of research included selection of engineering systems and building their structural as well as computational models. Particular attention was paid to choose appropriate representations which would enhance the search for novel design concepts. The results produced in this part are described in chapter 4.

- **Part Two – Implementation of the EED**

  This part consisted in developing a design support tool, named Emergent Designer, that implemented the proposed design method as well as representations of structural systems defined in part one. Emergent Designer is an integrated research and design support tool which applies models of complex systems to represent engineering systems and to analyze design processes. It is described in detail in chapter 5.

- **Part Three – Experimental Work**

  As stated in section 3.3, in this dissertation I focused on the core of the proposed design method, i.e. on phases 1 and 3 shown in Figure 15. Phase 1 is related to the research question 1 and phase 3 is linked to the research question 3 as it is shown in Figure 14. Hence, the two research questions have been further refined to address conceptual design problems in structural engineering. When the questions were precise enough, they formed the bases for hypotheses that could be tested empirically.

The third part of this dissertation included experimental work performed using Emergent Designer. The conducted experiments were directly related to the process of the empirical validation of Emergent Engineering Design which will be described in detail in the next section. They involved empirical validation of design concept generation mechanisms (chapter 6), evolutionary optimization mechanisms (chapter 7), and morphogenic evolutionary design, a combined approach in which design concept generation mechanisms (generative representations) were evolved by evolutionary algorithms (chapter 8).

- **Part Four – Analysis of the Experimental Results**

The results obtained in the conducted experiments have been analyzed both qualitatively (chapters 6 and 8) and quantitatively (chapters 7, 8). The qualitative analysis involved visual inspection of generated design concepts and a search for interesting and emergent structural shaping patterns. The quantitative analysis considered statistical properties of the design processes. Here, the performance of generated designs and efficiency of the design processes were investigated.

3.6. Validation Methodology

Emergent Engineering Design was validated using the Validation Square (Pedersen et al. 2000), a framework for validation of design methods, introduced in section 2.5.2. The validity of EED supports the claim of the *advancement of scientific knowledge* in the field of engineering design. The outline of the validation methodology used in this dissertation is presented in Figure 17. It is an extended version of Figure 12 in which all major stages of the validation process have been exemplified with specific tasks required to validate EED.

The bottom part of Figure 17 shows four elements of the Validation Square framework, i.e. Theoretical Structural Validity (TSV), Empirical Structural Validity (ESV), Empirical Performance Validity (EPV), and Theoretical Performance Validity (TPV). The four elements were used to evaluate effectiveness and efficiency of EED. TSV and ESV provided qualitative measures of EED's effectiveness, while EPV and TPV gave quantitative measures of EED's efficiency. Both qualitative and quantitative measures were utilized to demonstrate the usefulness of EED. The usefulness of EED supports the claim that the new scientific knowledge is *significant*. On the other hand, novelty of the proposed design method was justified based on the literature reviews presented in chapter 2 as well as a discussion on the open issues in engineering design offered in section 3.2.

A detailed description of the tasks involved to validate Emergent Engineering Design, broken down by each of the four elements of the Validation Square, is presented below.

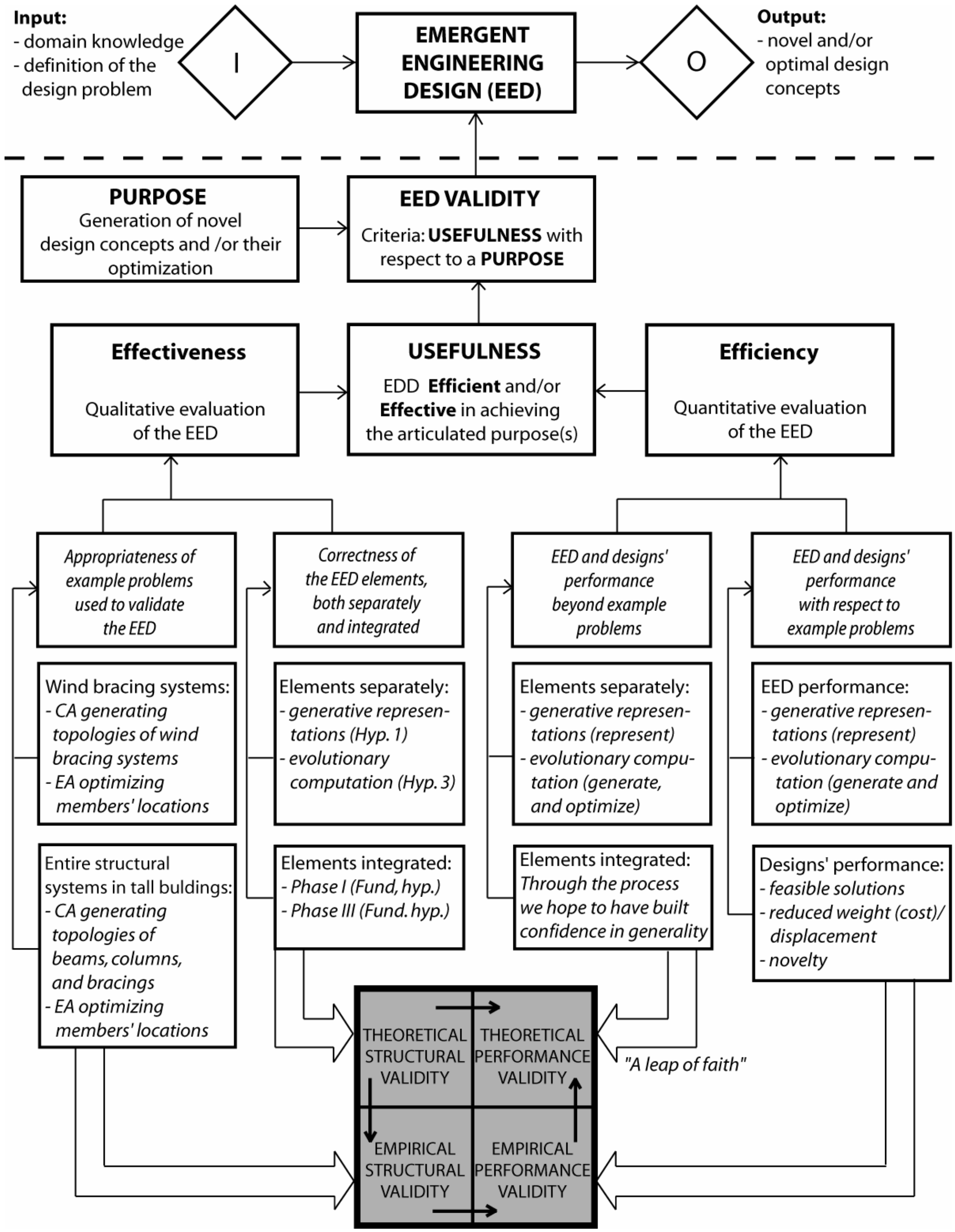


Figure 17. Outline of the validation methodology of Emergent Engineering Design using the Validation Square framework

3.6.1. Theoretical Structural Validation

Theoretical Structural Validity was supported by accepting the individual components constituting EED and accepting the internal consistency of the way the components were integrated. The validity of the individual components and their synthesis pertain to *structural soundness* of the proposed design method in a general, or theoretical, sense (Pedersen et al. 2000).

The confidence in validity of the individual components of EED was built based on the available scientific literature. It included evolutionary design and complex systems defining generative representations (see Phase I in Figure 14 and Hypothesis 1 in section 3.3), and evolutionary computation defining mechanisms of generation and optimization of design concepts (see Phase III in Figure 14 and Hypothesis 3 in section 3.3). One of the major goals of the literature review presented in chapter 2 was to establish confidence in validity of individual components of the proposed design method. On the other hand, the goal of chapter 4 was to establish confidence in validity of the way the components were integrated.

The confidence in *internal consistency* of EED was built using flow-chart representations of information flow within the proposed design method and within each of its components. It was demonstrated that for each component there was an adequate input available, that the anticipated output from the component was likely to occur based on the input and that the anticipated output is an adequate input for the next component. A detailed description of the flow of information in the proposed design method and in a computer system, called Emergent Designer, implementing the method is offered in chapter 5.

3.6.2. Empirical Structural Validation

Empirical Structural Validity of EED was supported by accepting the appropriateness of example problems that were used to test the proposed design method. The validity of the example problems pertains to the *empirical soundness* of the design method.

As stated in section 3.1, this dissertation investigates the proposed method only for conceptual design problems in structural engineering. Hence, two classes of conceptual design problems from the structural engineering domain were selected, namely design of a wind bracing system in a tall building and design of the entire steel structural system in a tall building. A brief description of the two classes of design problems and justification of their choice is offered in section 2.4.4.

The confidence in appropriateness of the example problems chosen to evaluate EED's performance was built by (Pedersen et al. 2000):

1. *Documenting that the example problems are similar to the problems for which EED's components are generally accepted.*
   SOTA overviews of all components of the proposed design method are included in chapter 2. The overviews discuss current research developments in these fields from the perspective of their relevance to engineering design. Moreover, each section in chapter 2 contains a subsection presenting structural engineering applications, if any, of the main ideas discussed there. In this way, the confidence in accepting the applicability of the components to the example problems was built.
2. *Documenting that the example problems represent the actual problems for which EED is intended.*
   The justification for the choice of the two example problems is presented in section 2.4.4. Moreover, chapter 4 demonstrates that the selected problems exhibit the

properties of problems for which EED is intended, e.g. they consist of a relatively large number of identical, simple, and locally interacting structural members.

3. *Documenting that the data associated with the example problems can support a conclusion*.
   As discussed in section 2.4.4, the example problems investigated in this dissertation are considered as one of the most complex and time-consuming design tasks in structural engineering. Therefore, they are of suitable complexity for the demonstration of the usefulness of the proposed design method.

Theoretical Structural Validity and Empirical Structural Validity qualitatively validate EED. The quantitative validation of the design method was tested by several performance measures which are described in the following sections.

3.6.3. Empirical Performance Validation

Empirical Performance Validity was supported by accepting that the outcome of EED is useful with respect to the initial purpose for the example problems. It was also accepted that the achieved usefulness was linked to the application of EED.

The purpose of the proposed design method for the example problems was directly related to the ultimate objective of this dissertation, i.e. generation of novel design concepts and their efficient optimization. The task of validating EED's performance empirically was divided into:

- two subtasks in which the empirical performance of the *individual components* of EED was measured for the example problems
- a third subtask in which the empirical performance of *integrated components* of EED was measured for the example problems

As stated in section 3.3, this dissertation investigated only the research questions 1 and 3 corresponding to phases 1 and 3 of the proposed design method. Hence, the first two subtasks tested the individual research hypotheses (the first subtask tested the research hypothesis 1 and the second subtask tested the research hypothesis 3) by measuring the performance of the corresponding component of EED, i.e. the generative representations component and the evolutionary computation component, for the example problems. The third subtask tested the fundamental hypothesis of this dissertation by measuring the performance of the integrated components of EED for the example problems.

The process of the empirical performance validation of EED was conducted in the following way. First, the usefulness of the generative representations component of EED in producing novel design concepts was tested empirically for the example problems (see chapter 6). In this subtask, no optimization algorithms were applied. Instead, various complex systems, modeled by simple programs, were used to explore the space of generative representations of structural systems. The produced design concepts were qualitatively and quantitatively compared to both randomly generated design concepts and to the best designs known from the structural engineering literature.

Second, the performance of the evolutionary computation component of EED was measured in optimizing the designs of engineering systems. Here, on the other hand, the emphasis was put on strictly optimization issues. Hence, the optimized engineering systems were represented by standard parameterized representations rather than by the generative ones (see chapter 7). Evolutionary-based optimization was initialized with a randomly generated initial population of solutions or with a set of design concepts that included state-of-the-art solutions known from the structural engineering literature. The design concepts optimized by the evolutionary computation component were compared to the solutions incorporated in the initial population of designs (thus, the improvement of the generated solutions was measured) and to the best designs known from the structural literature. The goodness/fitness of generated designs was measured in terms of the following evaluation criteria: the total weight of a steel structural system (which gives a reasonable estimate of the cost of a steel structure), and the maximum horizontal displacement of a steel structural system (which gives an estimate of the structure's stiffness).

Finally, the usefulness of the integrated components of EED was determined by measuring their performance both in producing novel design concepts and in their subsequent optimization. The obtained performance measures were compared to the results obtained using evolutionary-based methods utilizing standard parameterized representations which constitute the state-of-the-art in conceptual design of structural systems (topology optimization). Although traditional topology optimization methods based on linear programming techniques proved to be successful in finding optimal topologies for small design problems, they were rendered inadequate when the size of the problems considered was scaled up (see a detailed discussion presented earlier in section 2.1.7). Additional difficulties of traditional methods arise due to discontinuous nature of the design problems considered in this dissertation which was another reason for not including them in EED's empirical performance validation process. The obtained performance measures were analyzed statistically using appropriate statistical tests. Also, various experimental analyses and comparisons with state-of-the-art methods were conducted to demonstrate that the achieved usefulness was due to the application of EED (see chapter 8).

3.6.4. Theoretical Performance Validation

Theoretical Performance Validity was supported by accepting that the usefulness of EED extends beyond the example problems. The confidence in generality of the EED was built by *induction* that involved results from all previous validation steps, i.e. Theoretical Structural Validation, Empirical Structural Validation, and Empirical Performance Validation. The inductive argument was structured in the following way:

- Theoretical Structural Validity demonstrates that the individual components of EED are generally accepted for the applications the design method is intended. It also shows the internal consistency of EED.
- Empirical Structural Validity shows that the components of EED are applied within their accepted ranges.
- Empirical Performance Validity demonstrates the usefulness of EED for the example problems as well as that the usefulness is achieved due to the application of the method.

Based on that, the generality of EED was claimed, which is understood in this dissertation as its usefulness beyond the example problems (see chapter 9). Hence, Theoretical Performance Validity involved a 'leap of faith' to produce belief in a general usefulness of the proposed design method. The purpose of the previous steps in the Validation Square was to show 'circumstantial' evidence to facilitate this leap of faith (Pedersen et al. 2000).

3.7. Summary

In this chapter, I proposed and defined Emergent Engineering Design, the major objective of this dissertation, and described the structure of the argument presented in this dissertation. Initially, in the first section of this chapter I defined the problem considered in this dissertation, i.e. a need for a conceptually coherent method for designing engineering systems which addresses both important objectives of engineering design: development of novel designs and their optimization. In the following section, I showed that this problem is closely related to the open issues in the field of engineering design.

The third section defined Emergent Engineering Design, a design method based on models of complex systems and inspired by the processes occurring in nature. I presented the structure of the argument in the form of research questions and research hypotheses. I also described the scope of research conducted in this dissertation.

The fourth section of this chapter provided a detailed description of the assumptions incorporated in the proposed design method. First, an assumption of four phases of the conceptual design process (representation space definition, representation space decomposition, generation and optimization of design concepts, and fitness evaluation and adaptive testing) was debated and subsequently related to the corresponding phases in traditional design. Second, a discussion on the choice of the appropriate level of generality of representations of engineering systems (single structural elements or small sets of elements) was presented.

The fifth section outlined the conducted research while the sixth section of this chapter provided a detailed description of the validation methodology that was used to validate EED. The process of validation was based on the Validation Square framework (see section 2.5.2) and consisted of four major parts: Theoretical Structural Validation, Empirical Structural Validation, Empirical Performance Validation, and Theoretical Performance Validation. Each of them was described in detail and linked to the appropriate chapters of this dissertation.

4. DESIGN REPRESENTATIONS

"... the key element of design is representation. ... representation in design incorporates both representation of the artifact being designed as well as representation of the process by which the design is completed."

(Clive L. Dym)

In this chapter, I introduce computational representations of two classes of structural systems: wind bracing systems in tall buildings and entire steel structural systems in tall buildings. In doing that I conduct the first stage of the Theoretical Structural Validation process of EED (see section 3.6.1), in which I want to establish confidence in validity of the way the components of the proposed design method are integrated at the representational level. I will demonstrate that by first proposing several design concept generation mechanisms based on models of complex systems and then showing how these mechanisms can be encoded in the generative representations suitable for evolutionary optimization processes.

Figure 18 shows organization of this chapter. First, I present a general overview of representations of structural systems and a brief discussion on the level of their generality that is suitable for conceptual design. Next, I discuss traditional approaches to represent engineering systems in the form of parameterized representations of steel structural systems in tall buildings. At this point, I am ready to propose a new approach based on models of complex systems and inspired by the processes of *morphogenesis* occurring in nature. I define several types of design concept generation mechanisms based on cellular automata and discuss their computational and representational advantages and disadvantages. I also show how these mechanisms can be encoded as the generative representations which are suitable for evolutionary optimization processes.

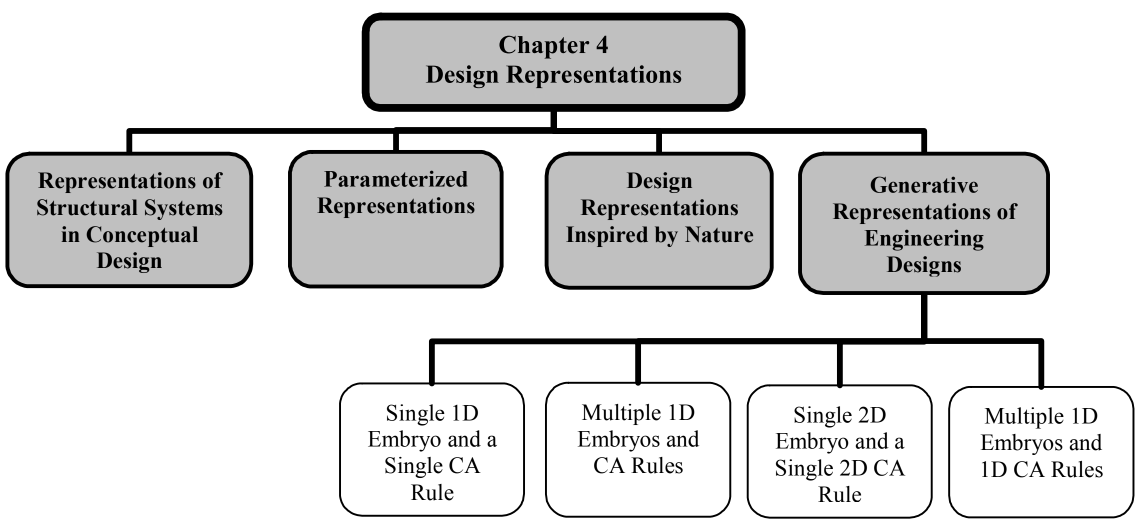


Figure 18. Organization of chapter 4

4.1. Representations of Structural Systems in Conceptual Design

As I discussed it earlier in section 2.1.3, representations in engineering design incorporate (Dym 1994):

- representation of an artifact (engineering system) being designed, and
- representation of a process by which the design is produced

In this chapter and in chapter 6, I focus on the new ways of representing engineering systems (artifacts) while in chapters 7-8 I introduce representations of design processes.

A *representation* of an engineering system can be defined as its computational description expressed in terms of attributes (Arciszewski et al. 1995). Attributes describing the system can be defined as a formal representation of its various characteristics including the structure's topology, its weight, etc. These attributes can be divided into two major groups: quantitative and qualitative. Quantitative attributes describe detailed characteristics of an engineering system that can be measured (Arciszewski 1988). These attributes are mostly considered in the analytical stage of a design process. On the other hand, qualitative attributes describe a general form of an engineering system and its characteristics that can not be explicitly measured, like shape, color, material used, etc. These attributes are usually multi-valued and take values from an unordered or partially ordered set of symbols (Arciszewski and De Jong 2001).

In general, a representation of an engineering system is a significantly broader description compared to its traditional model in engineering science because it encompasses much more knowledge than can be set into mathematical formulas and their numerical realizations (Dym 1994). This is particularly important in conceptual design, where most of the attributes describing a future engineering system are qualitative rather than quantitative and their selection involves significant amount of background knowledge. Hence, representations of structural systems in conceptual design usually consist of symbolic attributes.

In section 2.4.4, I introduced the two design problems considered in this dissertation to validate the proposed design method. They include the problem of designing a wind bracing system in a tall building and the problem of designing an entire steel structural system in a tall building. Both problems are examples of conceptual design problems and they can be naturally represented in terms of attributes taking appropriate set of *symbolic* values (see section 2.1.3).

A choice of a particular type of representation of an engineering system is highly influenced by a designer's goal, i.e. whether the emphasis is on optimality in terms of numerical values in the context of a specific design concept, or on generation of novel design concepts. In the former case, the attention is usually restricted to a particular concept or at most several concepts of existing designs and representations usually take a form of parameterizations. These types of representations have been traditionally used in evolutionary structural optimization (Kicinger et al. 2004b). I introduce parameterized representations of engineering systems in section 4.2. I will also investigate them experimentally in chapter 7 in which I empirically validate evolutionary optimization component of EED.

When a designer focuses on generation of novel design concepts, more general and usually more complex representations are used. Generative representations of engineering systems based on models of complex systems and inspired by the processes of morphogenesis occurring in nature are introduced in section 4.4. They are investigated experimentally in chapters 6 and 8.

4.2. Parameterized Representations

Parameterized representations are examples of direct representations (see section 2.1.3) in which each gene corresponds to an attribute encoding a dimension of the search space. Each such dimension represents an appropriate set of values, discrete, or continuous, which the attribute represented by this dimension can assume. As discussed earlier, in conceptual design, discrete values are usually preferred because they naturally encode symbolic values of attributes. In the simplest case, these representations use binary genes denoting the presence, or absence, of a feature. In such representations each individual consists of a fixed-length binary string of genes representing some subset of a given set of features. Often, in more complex engineering applications, as is the case with engineering design problems investigated in this dissertation, multi-valued attributes are used. In this dissertation representations of steel structural systems in tall buildings are encoded using integer-valued attributes. Parameterized representations described in this section are generalized versions of the encodings used in Inventor 2001 (Murawski et al. 2001).

In the design problems studied in this dissertation, a structural system of a tall building is considered as a system of identical parallel planar transverse structures, which are the subject of design. The representation space has been developed using the concept of division of the structural grid of the building (the system of vertical and horizontal lines of columns and beams, respectively) into units, or cells. A *cell* can be described as a part of the structural grid contained within the adjacent vertical and horizontal grid lines (Murawski et al. 2001).

Representations of steel structural systems in tall buildings considered in this dissertation encode the following types of structural members: bracings, beams, columns, and supports. Depending on the investigated design problem, either only a subset of the structural members or the entire set of all structural members was considered. Thus, in the wind bracing system design problem only bracings were used. In this case, all other structural members were assumed the same during the entire design process. On the other hand, in the problem of designing the entire steel structural system in a tall building, all structural members, including bracings, beams, columns, and supports, were considered and subjected to changes.

Figure 19 shows the values of the attributes representing wind bracing elements in a steel structural system at a phenotypic, symbolic, and genotypic level. Each such attribute can have up to seven symbolic values (see Figure 19b)) encoding various types of bracings (no bracing, diagonal bracing \, diagonal bracing /, K bracing, V bracing, simple X bracing, and X bracing). Their phenotypic, or design, representation is presented in Figure 19a). Figure 19c) shows genotypic values of the attributes representing bracing elements where alleles take on subsequent integer values from 0 to 6.

In Figure 20, phenotypic (see Figure 20a)), symbolic (see Figure 20b)), and genotypic (see Figure 20c)) values of attributes representing beam elements are presented. Each attribute representing a beam in a steel structural system can have up to five symbolic values encoding various types of beams (no beam, pinned-pinned beam, fixed-fixed beam, pinned-fixed beam, and fixed-pinned beam). The phenotypic (design) representation of beam attributes is presented in Figure 20a) while their genotypic values taking on subsequent integer values from 0 to 4 are shown in Figure 20c).

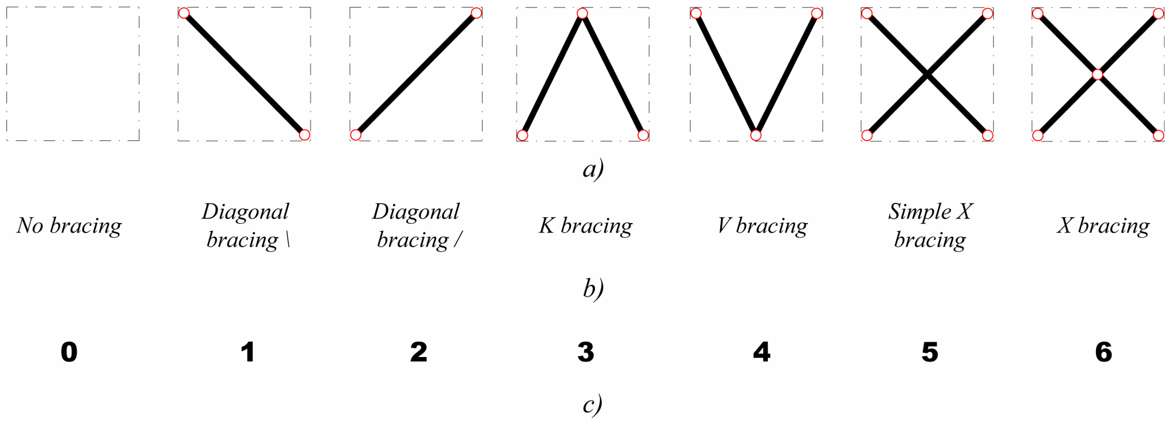


Figure 19. Values of the attributes describing bracing elements a) phenotypic representation, b) symbolic representation, c) genotypic representation

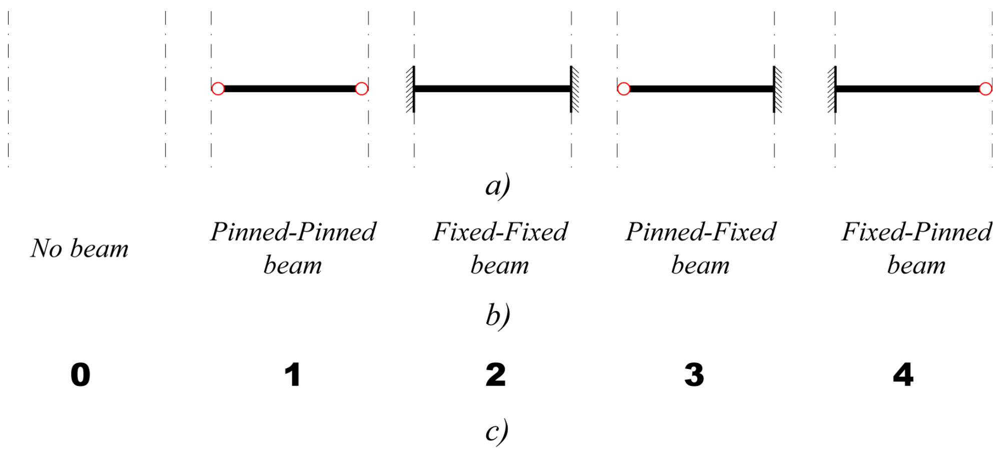
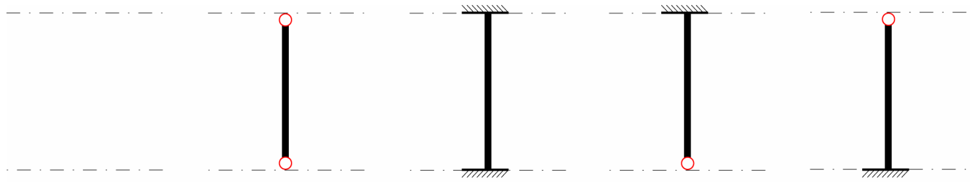


Figure 20. Values of the attributes describing beam elements a) phenotypic representation, b) symbolic representation, c) genotypic representation

Phenotypic, symbolic, and genotypic values of attributes representing column elements are presented in Figure 21. Here, similar to beam attributes, up to five symbolic values (see Figure 21b)) encoding various types of columns (no column, pinned-pinned column, fixed-fixed column, pinned-fixed column, and fixed-pinned column) can be used. Design representation of column attributes is presented in Figure 21a) while their genotypic values taking on subsequent integer values from 0 to 4 are shown in Figure 21c).

Finally, Figure 22 shows phenotypic, symbolic, and genotypic values of attributes representing supports. Here, four possible types of supports are allowed and encoded by four symbolic values (no support, pinned support, fixed support, and roller support) as it is shown in Figure 22b). Design representation of support attributes is presented in Figure 22a) and their genotypic values ranging from 0 to 3 are shown in Figure 22c).



a)

No column | *Pinned-Pinned column* | *Fixed-Fixed column* | *Pinned-Fixed column* | *Fixed-Pinned column*

b)

0 **1** **2** **3** **4**

c)

Figure 21. Values of the attributes describing column elements a) phenotypic representation, b) symbolic representation, c) genotypic representation

a)

No support | *Pinned support* | *Fixed support* | *Roller support*

b)

0 **1** **2** **3**

c)

Figure 22. Values of the attributes describing supports a) phenotypic representation, b) symbolic representation, c) genotypic representation

When parameterized representations are employed, a given structural system can be encoded as a sum of representations of its individual cells, each described an attribute identifying the existence and the type of a structural member. Figure 23 shows a simple example of this approach. 10-story building with 4 bays is divided into 40 cells contained within the adjacent vertical and horizontal grid lines (see Figure 23a)). In this problem, only wind bracing elements are the subject of design and all other structural elements of the steel structure, i.e. beams, columns, and supports, are assumed the same during the entire design process. Figure 23a) shows a configuration of bracing elements in a steel structure which represents a design concept. The representation of a design concept at this level is called a phenotypic representation, or simply a phenotype. The phenotype is created (decoded) from a genotypic representation involving 40 integer-valued attributes that represent 40 bracing elements in the steel structure as it is shown in Figure 23b).

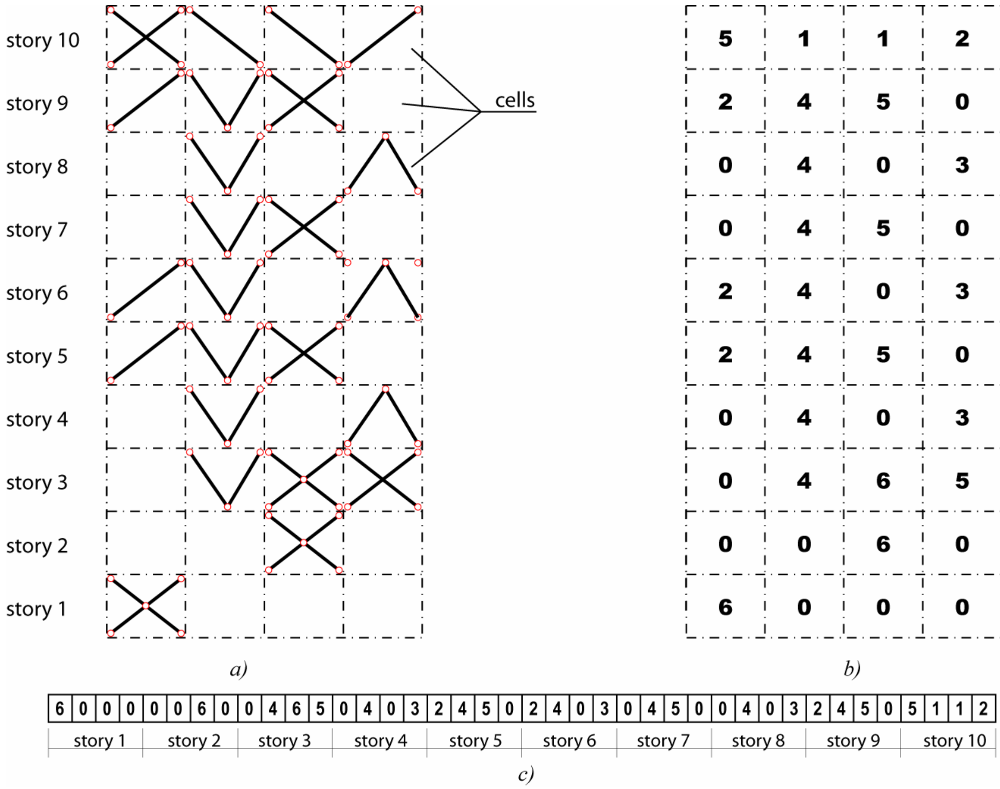


Figure 23. a) Phenotypic representation of a wind bracing system, b) The same system represented by multi-valued integer attributes, c) Linear genome representation of the system that is manipulated by an evolutionary algorithm

Usually, the actual genotypic representation, or genome, that is manipulated by an evolutionary algorithm, is linearized and encoded as a string. In this particular example, it is a string of integer values (see Figure 23c)). All genotypic representations considered in this dissertation are linear. A collection of all such genotypes forms a *genotypic space* of the domain. A collection of all phenotypes corresponding to all combinations of attribute values (all genotypes), forms a *phenotypic space* of the domain.

As is it discussed above and shown in Figure 23, in the design problems considered in this dissertation a clear distinction between genotypic and phenotypic spaces is made. Evolutionary search operates in the genotypic space but fitness evaluation is performed in the phenotypic space.

As mentioned earlier, the two design problems investigated in this dissertation include conceptual design of wind bracing systems in tall buildings, and conceptual design of the entire steel structures in tall buildings. In the former case, the subject of design is the placement and type of bracing elements only. In the latter case, design involves the placement and type of all structural elements discussed earlier, i.e. beams, bracings, columns, and supports.

In this dissertation, fixed-length genotypes are used as representations of various steel structural systems. The length of a genotype used in a given situation, however, depends on the design problem being studied (wind bracing system or the entire steel structural system) and on the number of cells in the structural system being considered. The number of cells is obviously related to the number of stories and the number of bays in a tall building. Once the design problem and topological properties of a tall building are determined, then the length of the genotype is completely defined and does not change.

When the parameterized representations are used, the lengths of genotypes for the wind bracing system design problem are simply equal to the number of cells in a given tall building. For example, in the simple design problem described earlier and shown in Figure 23, the genome representing a parameterized design concept of a 10-story building with 4 bays has the length of 40 genes (bracing attributes). The situation is more complicated when design of the entire steel structure is considered. In this case, all structural elements, including beams, bracings, columns, and supports are represented. Figure 24 shows the same 10-story building with 4 bays for which the entire steel structural system is being designed.

Figure 24a) shows a configuration of a steel structure in a tall building in which all structural elements are represented. This configuration forms a phenotype of a particular design concept. The phenotype is created (decoded) from a genotypic representation involving 40 integer-valued attributes that represent 40 bracing elements, 40 integer-valued attributes that represent 40 beam elements, 50 integer-valued attributes that represent 50 column elements, and 5 integer-valued attributes that represent 5 supports in the steel structure (see Figure 24b)). Hence, the total number of genes (attributes) that completely define this steel structural system in this parameterized representation is equal to 135. The structure of the actual linear genome manipulated by an EA and consisting of 135 genes is presented in Figure 24c).

Emergent Designer, which will be introduced in chapter 5, allows for a choice of:

- Structural elements considered in design,
- Values of attributes defining types of structural elements.

Thus, one can, for example, consider only bracings and beams and assume all columns and supports the same during the entire design process. Similarly, one can choose, for instance, that during the design process the only allowed values for beam attributes are pinned-pinned beams and fixed-fixed beams.

4.3. Design Representations Inspired by Nature

Parameterized representations discussed in the previous section have been widely used in engineering optimization. This dissertation, however, emphasizes **both** novelty and optimality in engineering design. To achieve the ultimate objective of this dissertation, i.e. generation of novel design concepts and their efficient optimization, other types of representations of engineering systems had to be proposed. The inspiration for the design representations introduced in this section comes again from nature which manipulates *rules* for growing complex organisms, called 'genetic plans', rather than the complex organisms themselves. The organisms are then built from the plans via a developmental process called morphogenesis (Thompson 1942). *Morphogenesis* can be described in several ways, including the following 3 definitions (adapted from *Principia Cybernetica*):

Definition 1. ***Morphogenesis***

"Morphogenesis is an evolutionary development of the structure of an organism or a part."

Definition 2. ***Morphogenesis***

"Morphogenesis is an embryological development of the structure of an organism or a part."

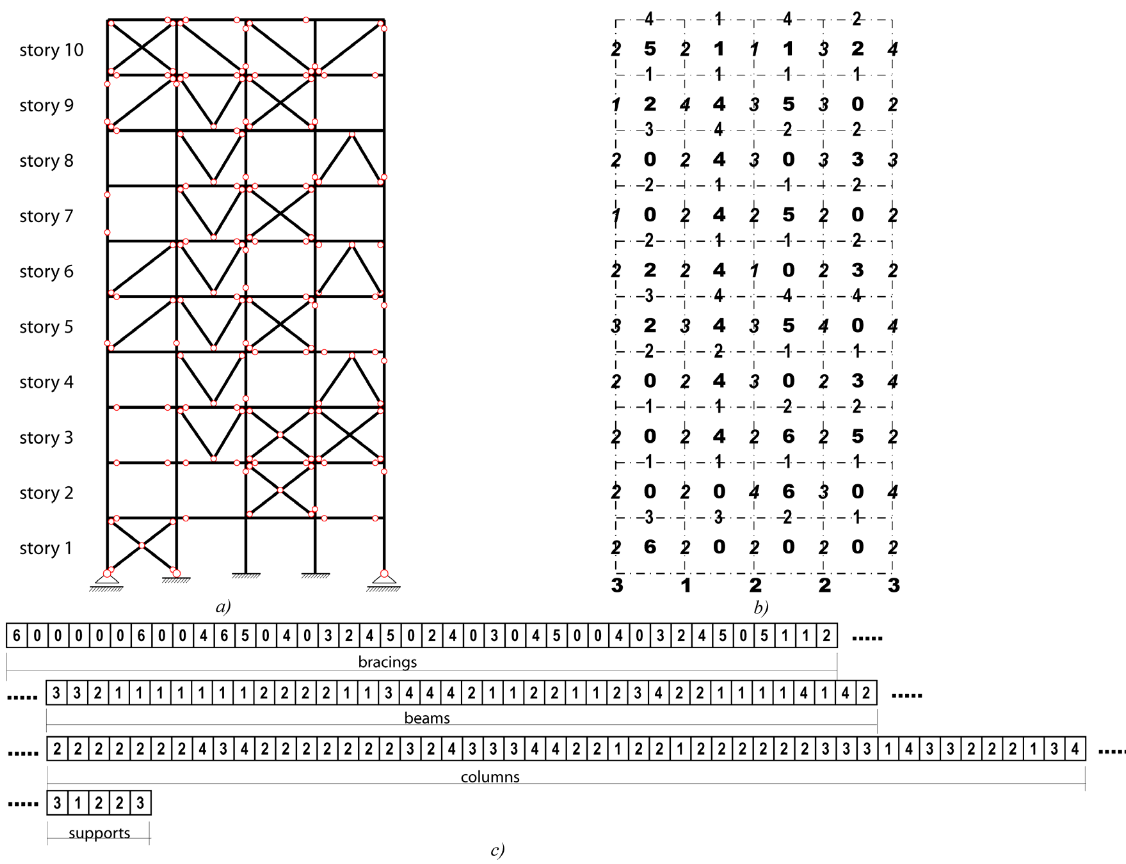


Figure 24. a) Phenotypic representation of an entire steel structural system, b) The same system represented by multi-valued integer attributes, c) Linear genome representation of the system that is manipulated by an evolutionary algorithm

Definition 3. ***Morphogenesis***

"Morphogenesis is the process in complex system-environment exchanges that tends to elaborate a system's given form or structure. Examples are the growth of an animal from a fertilized ovum, biological evolution, learning, and societal development. A morphogenic system is capable of maintaining its continuity and integrity by changing essential aspects of its structure or organization."

The definitions 1 and 2 are most closely related to the ideas presented in this dissertation.

Due to the lack of appropriate terminology describing the use of generative representations within the field of engineering design, this dissertation introduces a new term *morphogenic evolutionary design*. By combining the definitions of morphogenesis and engineering design (introduced earlier in section 2.4.1), I can define morphogenic engineering design in the following way.

Definition 4. ***Morphogenic Evolutionary Design***[1]

Morphogenic evolutionary design is the systematic generation and evaluation of representations of engineering systems or their parts whose form and function achieve stated objectives and

[1] I would like to thank Prof. Tomasz Arciszewski and Dr. Sanjeev Kumar for their comments and suggestions that helped me improve this definition.

satisfy specified constraints. It is done using the mechanisms inspired by the processes of developmental biology and evolution.

One of the key aspects of morphogenic engineering design is representation of an engineering system being designed. Recently, several researchers investigated the potential of using indirect and generative representations inspired by the processes of morphogenesis in creative design (Bentley and Kumar 1999; Hornby 2003). As discussed in section 2.1.3, indirect representations do not encode complete design concepts, as do parameterized representations, but rather *rules on how to develop*, or *grow*, these designs. Generative representations are examples of indirect representations that can *reuse some parts of an encoded design* during the phenotype construction phase. Their ability to reuse elements of an encoded design improves the search efficiency in large design spaces as well as scalability by capturing design dependencies (Hornby 2003).

Figure 25 illustrates the concept of design inspired by nature in the context of designing steel structural systems in tall buildings. Similarly as in nature (see the bottom part of Figure 25), a building is developed from an initial seed (called here the design embryo) and then 'grown' to its fully-developed form. This process, called morphogenic design, is inspired by the processes of morphogenesis occurring in nature, e.g. in the process of development of plants.

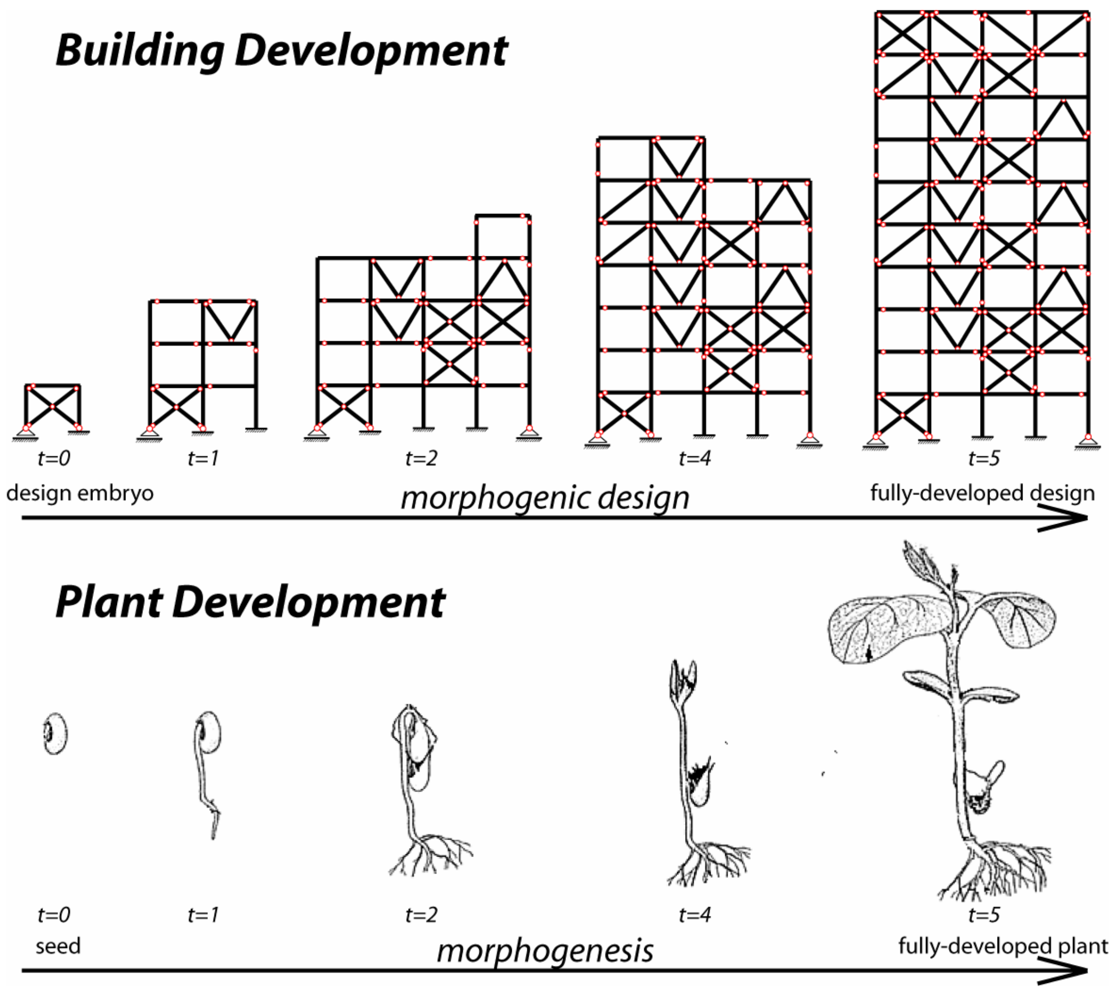


Figure 25. Process of development of a steel structural system in a tall building inspired by the processes of morphogenesis occuring in nature

Next section further extends the ideas presented here and proposes specific examples of generative representations of steel structures in tall buildings based on various models of complex systems, including one-, and two-dimensional cellular automata. It also proposes the generative representations of the entire steel structural systems in tall buildings.

4.4. Generative Representations of Engineering Designs

In this section, I propose and define generative representations of the two design problems considered in this dissertation. These representations are based on models of complex systems. The generative representations proposed here, and inspired by the processes of biological development, consist of two parts: encoding of a 'design embryo' and encoding of a 'design rule,' which is applied to the design embryo to develop a design concept from it.

A *design embryo* is understood in this dissertation as an ordered set of cell values representing an initial configuration (one-, or two- dimensional) of structural members (e.g. wind bracing types) from which a design concept is developed.

A *design rule*, on the other hand, is a formal description of a transformation that changes the current configuration of structural members into a new configuration. This transformation defines a unit time step. In this dissertation, various types of one- and two-dimensional cellular automata are considered as representations of the design rules. Thus, it is possible to provide a more specific definition of a design rule in this context. It is defined as a systematic definition of a transformation that updates the current configuration of cell values (representing the corresponding types of structural members) into a new configuration of cells at a subsequent time step. This transformation consists of three major components:

- A complete set of decision rules whose conditions incorporate all possible combinations of cell values (types of structural members) in the given local neighborhoods and their outcomes specify the values of the central cells of these neighborhoods at a next time step,
- Assigned sequence/ordering of the individual decision rules, which is assumed the same for the entire class of the design rules and hence can uniquely define every design rule belonging to this class,
- A complete set of outcomes associated with individual decision rules and having the same ordering as for the decision rules.

The genomes encoding the generative representations proposed in this dissertation have, in general, the following structure. The first part of each genome encodes the design embryo while the second part encodes the corresponding design rule.

The following sections introduce several types of generative representations and instantiate the ideas presented above. They also provide detailed descriptions of the processes of development of design concepts using various kinds of design embryos and the corresponding design rules.

4.4.1. Single 1D Embryo and CA Rule Representing Wind Bracings

One of the simplest instances of generative representations proposed in this dissertation consists of a design embryo formed by a single one-dimensional initial configuration of cells and a design rule represented by a single 1D CA rule. This rule is applied to the design embryo and develops a design concept of a wind bracing system in a tall building. In this case, the design embryo is the configuration of the first story in a wind bracing system of a tall building. The design rule is applied to the design embryo and iterated for the number of times that is one less than the number of stories in a tall building.

The process of applying this type of generative representation to develop, or grow, a design concept of a wind bracing system is illustrated graphically in Figure 26. First, Figure 26a) shows the process of iteration of an elementary CA. In this case, the individual cell states have only binary values and local neighborhoods affecting the iteration of a considered cell are formed by this cell and its immediate left and right neighbors. Therefore, groups of three cells are considered in each local neighborhood and such situation is called a 'local neighborhood of size 3'. The bottom row of Figure 26a) consists of 6 squares (cells) denoting an initial configuration of cells (*t=0*). In this particular case, the initial configuration consists of cell state values 0 0 0 1 1 0. White squares in Figure 26 denote cell state values equal to 0 while black squares represents cell state values equal to 1. A graphical representation of the particular CA rule used to iterate this initial configuration for 15 time steps is presented in Figure 26b). As discussed earlier, a CA rule can be understood in the context of this dissertation as a complete set of decision rules whose conditions incorporate all possible combinations of cell state values in a given local neighborhoods (here of size 3) and the outcomes determine the values of the considered cells (usually central cells in a local neighborhood) at the next time step. If the ordering of the individual decision rules shown in Figure 26b) is assumed the same, then any CA rule can be uniquely defined by the outcome values (the top row in Figure 26b)) associated with individual decision rules.

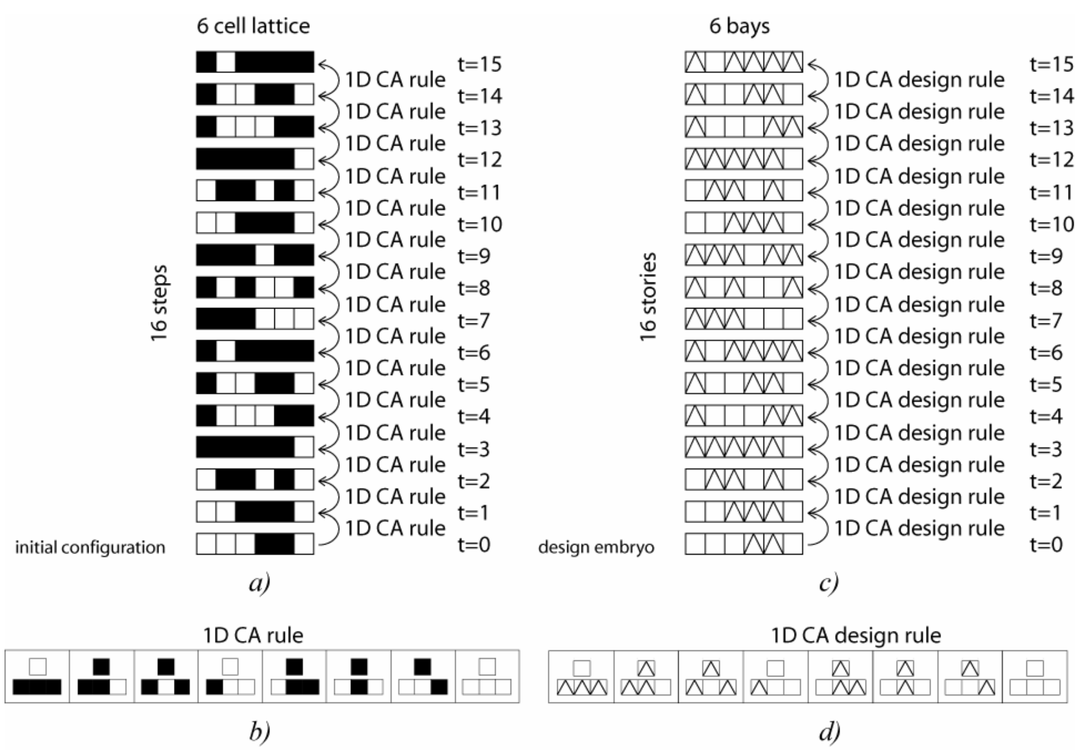


Figure 26. a) Process of iteration of a 1D CA starting with an initial configuration consisting of 6 cells, b) Graphical representation of a 1D CA rule assigning values to a central cell in a local neighborhood (the top row) at a next time step (the bottom row), c) Process of generation of a wind bracing design concept from a design embryo using a 1D CA design rule, d) Graphical representation of a design rule based on a 1D CA which assigns values to a central cell in a local neighborhood (the top row) at a next time step (the bottom row)

A design concept of a wind bracing system is created analogically. Figure 26c) shows a process of development of a design concept from its design embryo (the initial configuration of bracings at the first floor) using a design rule presented in Figure 26d). The design rule represented by a 1D CA can be thought of as a complete set of decision rules whose conditions (the bottom part of Figure 26d)) incorporate all possible combinations of types of bracings in the given local neighborhoods and the outcomes specify the values of the central cells of these neighborhoods at a next time step (the top part of Figure 26d)). In this case, only two values of the attributes representing bracing elements are used: no bracing (empty cell) and K bracing. The design embryo forms the first story in the generated design concept (*t=0*). As it is shown in Figure 26c), the 1D CA design rule is iterated the number of times that is one less than the number of stories in a tall building. The process starts at the bottom level and gradually moves upwards. This choice, however, is arbitrary and other starting conditions can be specified, e.g. a design concept can be built downwards starting from the design embryo located at the top level.

Figure 26 illustrates an incremental mechanism of generation of design concepts using a design embryo and a 1D CA design rule. It does not explicitly show, however, how subsequent configurations of stories are obtained, or in other words, it does not show how a 1D CA works. That is presented in Figure 27 which demonstrates the process of determining the configurations of cells at subsequent time steps in more detail.

Figure 27b) shows the same 1D CA rule as in Figure 26b) that is applied to the same initial configuration of cells as in Figure 26a). The process of generation of subsequent configurations at time steps *t=1, 2, 3 …* is presented graphically in Figure 27a). First, a set of local neighborhoods of size 3 (it is an elementary CA) is constructed by taking each cell from the initial configuration together with its left and right neighbors and placing them respectively in the middle, left, and right of the lattice defining each local neighborhood (see the set of 6 local neighborhoods of size 3 placed above the initial configuration in Figure 27a)). In this particular example, so-called cyclic (or periodic) boundary conditions are used, meaning that the rightmost cell in the initial configuration becomes the left neighbor of the leftmost cell in the initial configuration, and vice versa (denoted by dashed lines in Figure 27a)).

Next, the local neighborhoods created that way are compared to the local neighborhoods shown in the bottom row of Figure 27b). When the two match, the value shown in the top row of Figure 27b) defines the new value of the central cell at the next time step. This process is repeated for each local neighborhood and the values obtained are placed in appropriate positions in the new configuration of cells at time *t=1,* thus completely defining this configuration. The process is repeated for an arbitrary number of steps. Figure 26a) shows the results of the iteration process for the first 15 steps. Figure 27a) gives a detailed representation of the process of determining the new configuration at a subsequent time step of the first 3 iterations only.

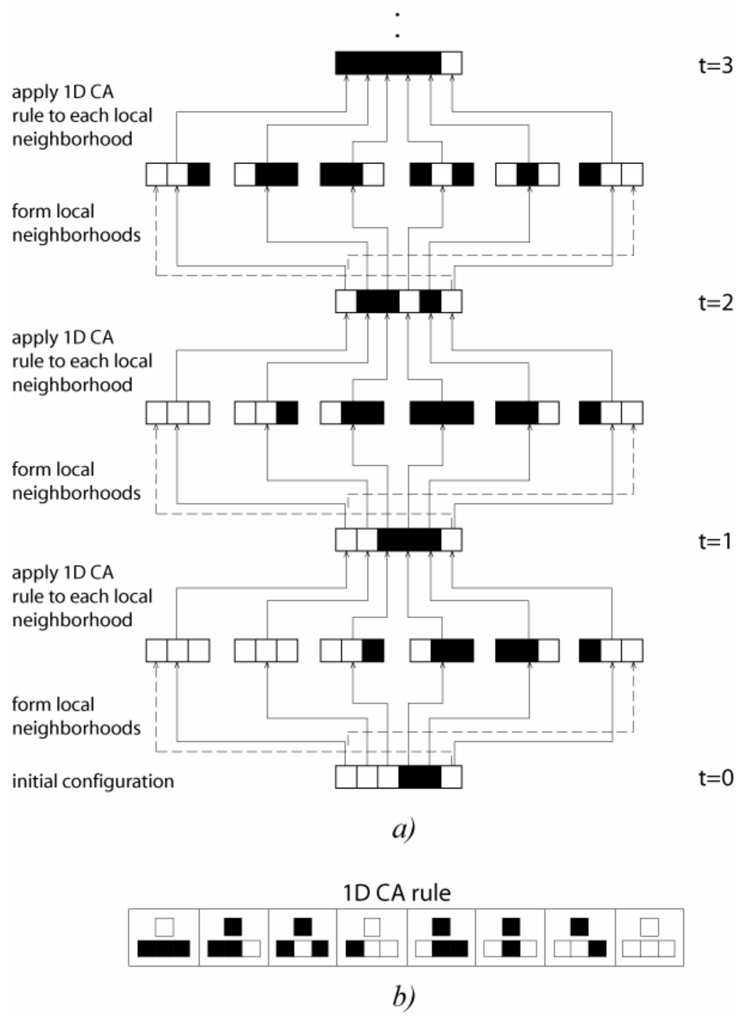


Figure 27. a) Generation of subsequent configurations (t=1, 2, 3, ..) of the lattice of cells during the process of iteration of a 1D CA starting with an initial configuration (t=0) consisting of 6 cells, b) Graphical representation of a 1D CA rule used in part a)

Representations defined in this section can be used as design concept generators (see chapter 6). They can be also evolved using evolutionary algorithms (see chapter 8). In this case, however, a design embryo and a design rule used to develop a design concept of a wind bracing system have to be appropriately encoded in a genome that is manipulated by an evolutionary algorithm. Figure 28a) shows a schematic view of the structure of such genome. It consists of two parts: a design embryo encoded in the first part of the genome (gray cells) and a design rule occupying the second part of the genome (white cells). The number of genes encoding the design embryo is equal to the number of bays in a tall building. On the other hand, the number of genes encoding the design rule depends on the number of possible cell state values (types of bracing elements) and the size of the local neighborhood. In order to encode a design rule in a genome in this way, one also has to assume an ordering of the individual decision rules making up the design rule. This ordering must be the same for the entire class of the design rules so that every design rule belonging to this class can be uniquely defined.

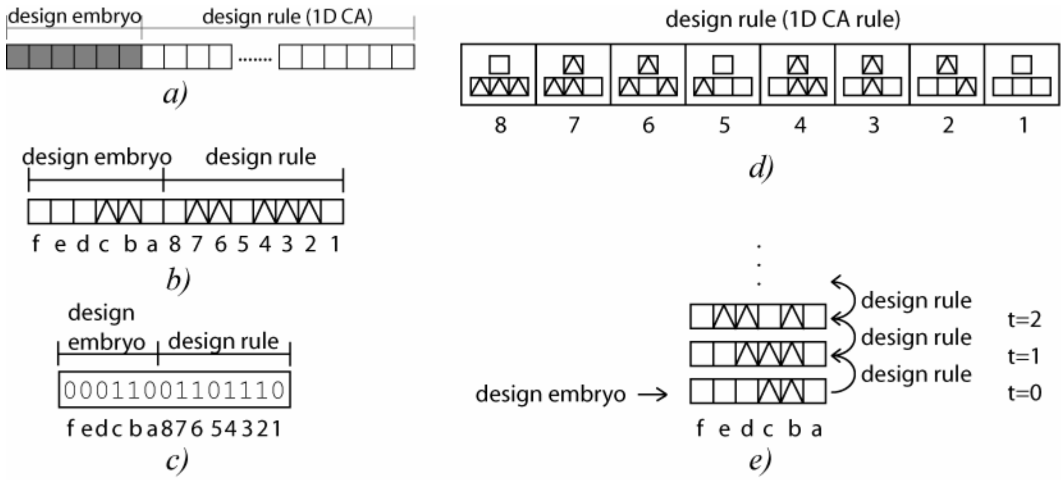


Figure 28. a) Schematic view of the structure of a genome encoding the generative representation of a wind bracing system consisting of a single design embryo (gray cells) and a single design rule (white cells), b) graphical illustration of an encoding of a design concept using the design embryo shown in part e) and the design rule shown in part d), c) encoding of the same design concept as in part b) but in an actual numerical form that is manipulated by an evolutionary algorithm

Figure 28b) graphically illustrates the genome encoding a design concept of a wind bracing systems presented earlier in Figure 26c). Here, the design embryo is represented by genes a-f in Figure 28b). As it is shown in Figure 28e), the design embryo encoded in the first part of the genome defines the configuration of the first story in a wind bracing system of a tall building (cells a-f at $t=0$). This choice, however, is arbitrary and other starting configurations can be used, e.g., the design embryo located at the top of a tall building.

The design rule, encoded in the second part of the genome (genes 1-8 in Figure 28b)), is represented by a 1D CA rule (see Figure 28d)). In this case, the design rule uses only two possible cell state values (empty cell denoting no bracing, and non-empty cell denoting K bracing), and the neighborhood of size 3. As defined earlier, it consists of a complete set of decision rules whose conditions (the bottom part of Figure 28d)) incorporate all possible combinations of types of bracings in the given local neighborhoods and the outcomes specify the values of the central cells of these neighborhoods at a next time step (the top part of Figure 28d)). All possible combinations of conditions of the design rule (see Figure 28d)) are ordered from 1 to 8. If this ordering is assumed the same for the entire class of the design rules with binary cell state values and the local neighborhood of size 3, then the outcome values (shown in the top part of Figure 28d)) uniquely define every rule belonging to this class. This important feature has been used in the definition of the encoding of the design rule in the genome shown in Figure 28b). Here, genes 1-8 encode the outcome values produced by the design rule presented in Figure 28d) and, given the assumed ordering, uniquely define it.

The genome encoding the representations of wind bracing systems in tall buildings described in this section consists of homogeneous genes representing integer-valued attributes of bracing elements. As it was shown in Figure 19, the attributes representing types of bracing elements can have up to 7 values.

The advantages of this representation include compactness and excellent scalability. A genome encoding a wind bracing system shown in Figure 26c) is 14 genes long when 2 cell state values (types of wind bracing elements) are used and 349 genes long when 7 cell state values are used. The representation can be even more compact when the design rule is represented by a 1D totalistic CA rule. In a totalistic CA the value of the current cell at the next time step depends on the *average* value of cells in the local neighborhood, and not on their individual values. Figure 6 (page 38) illustrates the process of iteration of a totalistic CA with three cell state values (see Figure 6a)) and the structure of a corresponding totalistic CA rule used in this process (see Figure 6b)). If the design rule is represented by a totalistic CA rule, then the genome encoding the design concept of a wind bracing system is 10 genes long when 2 types of bracings are used and 25 genes long when all 7 types of bracings are utilized.

The disadvantage of this approach is that a single design rule is applied at each story of a wind bracing system and hence it is impossible to diversify design rules for various parts of the wind bracing system, e.g. in traditional design different design rules may be used in the bottom part of the structure, where internal forces are the largest, compared to the upper part of the structure where internal forces are the smallest but local stiffness requirements are the same.

4.4.2. Multiple 1D Embryos and CA Rules Representing Wind Bracings

One of the limitations of the generative representations described in the previous section is the lack of diversification of design rules for various parts of a wind bracing system. Generative representations introduced in this section are aimed to remedy this problem. These representations consist of a *set* of one-dimensional initial configurations of cells and a *set* of 1D CA rules. Similar to the representations introduced in the previous section, each initial configuration, or design embryo, consists of a lattice of cells whose length is equal to the number of bays in a tall building. This representation also assumes that both the number of design embryos and the number of corresponding design rules (represented by 1D CA rules) are equal to the number of stories in a tall building. Figure 29 shows how a design concept of a wind bracing system is developed using this type of generative representations. In this case, a concept of a wind bracing system for a 16-story tall building is developed from 16 design embryos and 16 design rules (1D CA rules). In this particular example, only 3 cell state values (denoted by 3 colors: white, gray, and black) are used and correspond to three types of wind bracing elements, e.g. no bracing (empty cell), K bracing, and X bracing.

Each design rule is applied to its own design embryo, e.g., design rule 1 is applied to design embryo 1, design rule 2 to design embryo 2, etc., and iterated an arbitrary number of times, denoted in Figure 29 by *iteration_max*. Thus, the number of iterations of design rules (*iteration_max*) becomes an additional parameter for this representation. The iteration of all design rules is performed synchronously. The final configuration obtained during this process, i.e. configuration at *t=iteration_max*, forms a design concept which is subsequently evaluated.

Figure 30 shows a schematic view of the structure of the genome manipulated by an evolutionary algorithm. The genome consists of twice as many parts as the number of stories in a tall building. Gray cells in Figure 30 encode design embryos for various stories, while white cells encode design rules applied to these embryos. Similar to the representation introduced in the previous section, the number of genes defining any design embryo depends only on the number of bays in a tall building. Also, the number of genes encoding a design rule represented by a 1D CA rule is determined by the number of possible cell state values (types of bracing elements) and by the size of the local neighborhood. The encoding of all design rules in the genome is analogical to the one described in the previous section. The genome consists of homogeneous genes representing integer-valued attributes.

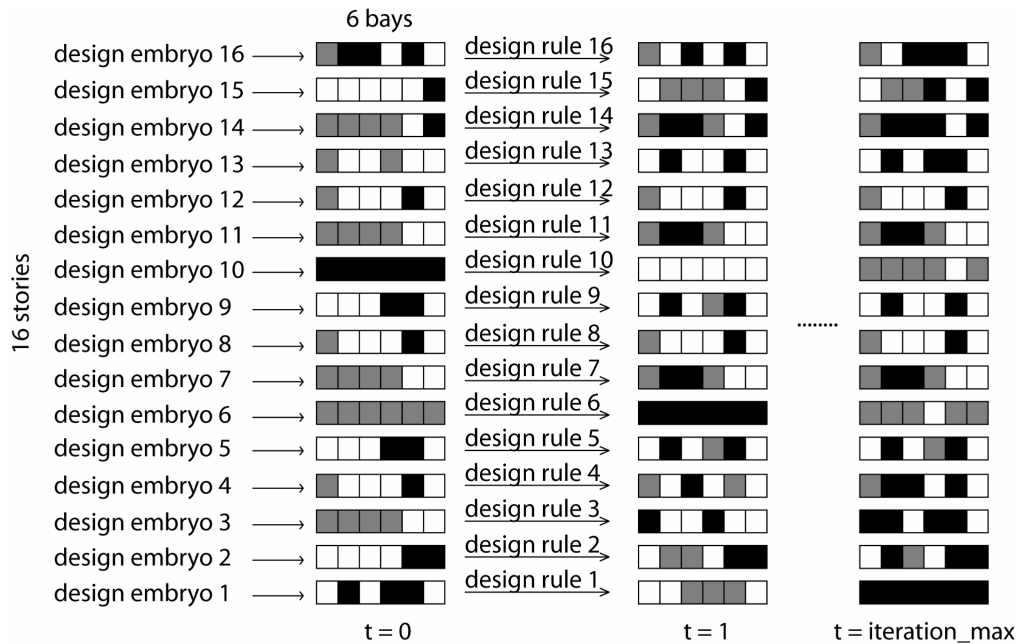


Figure 29. Process of generation of a design concept of a wind bracing system from a set of design embryos and a set of design rules

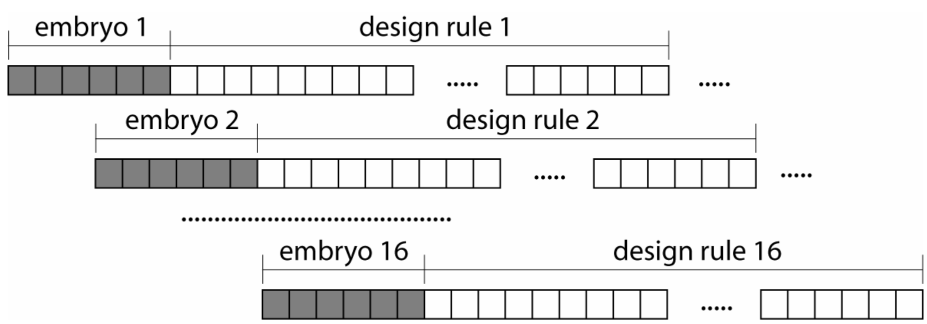


Figure 30. Schematic view of the structure of a genome encoding the generative representation of a wind bracing system and consisting of multiple design embryos (gray cells) and multiple design rules (white cells)

The advantages of this type of representation include mentioned earlier diversification of the design rules in various parts of the wind bracing system. The main disadvantage of this representation is the fact that it does not scale well with neither the number of cell state values (types of bracing elements) nor with the number of stories and bays in a tall building. The genome encoding the wind bracing system shown in Figure 29 consists of the following number of genes:

- Each design embryo is represented by a string of ternary values consisting of 6 genes,
- Each design rule is represented by a string of ternary values consisting of 27 genes (when a 1D CA rule with 3 cell state values and the local neighborhood of size three is used).

Hence, the entire genome of a 16-story building is encoded as a string of 528 genes. Increasing the number of cell state values to 7 would lengthen the genome to 5,584 genes. In this case, a more feasible approach involves design rules represented by 1D totalistic CA rules (see a brief discussion of totalistic 1D CA rules in the previous section and on page 38). Then, the length of the genome would be equal to 208 and 400, when 3 and 7 cell state values are used, respectively.

Generative representations described in this and previous sections form two extreme cases of representations of wind bracing systems involving one-dimensional embryos and the design rules represented by 1D CA rules. The generative representation introduced in the previous section develops the entire design concept of a wind bracing system from a single design embryo and a single design rule. On the other hand, this section defines the representation with the maximum possible (limited by the number of stories in a tall building) number of design embryos and design rules. One can, of course, easily define generative representations located somewhere in between the two extremes, i.e., a design concept of a wind bracing system can be developed from, for example, three design embryos and three design rules. In such a case, the first embryo and the first design rule would develop the bottom part of the structure, where the internal forces are the largest, the second embryo and rule would build the middle part of the structure, and finally the last embryo and rule would generate the upper part of the structure where internal forces are the smallest. Such representations would constitute a knowledge-driven engineering design in which available background knowledge on the design problem is incorporated in the representation of the considered engineering system.

4.4.3. Single 2D Embryo and 2D CA Rule Representing Wind Bracings

Representations of a wind bracing system proposed in this section are based on two-dimensional cellular automata (2D CAs). Here, a design rule is represented by a 2D CA and acts upon a design embryo which is now a two-dimensional array. This array represents an initial configuration of an entire wind bracing system. Figure 31 shows a process of developing a design concept of a wind bracing system using this generative representation.

A design embryo, in the form of a 2D array, is iterated an arbitrary number of times (*iteration_max* times) using a design rule represented by a 2D CA rule. Here, similar to the representation introduced in the previous section, the number of iteration steps (*iteration_max*) becomes an additional parameter that needs to be defined. Moreover, one more parameter has to be defined for this representation, namely the shape of the local neighborhood, to completely define the design rule. The two most popular and frequently used shapes of the local neighborhood include von Neumann neighborhood (von Neumann 1966) and Moore neighborhood (Moore 1962) (see a detailed description of these neighborhoods in section 2.2.1). Several other shapes of local neighborhood were considered in this dissertation. They will be described in chapter 6. The final configuration obtained during this process of iteration, i.e. configuration at the time step *t=iteration_max*, defines a design concept which is subsequently evaluated.

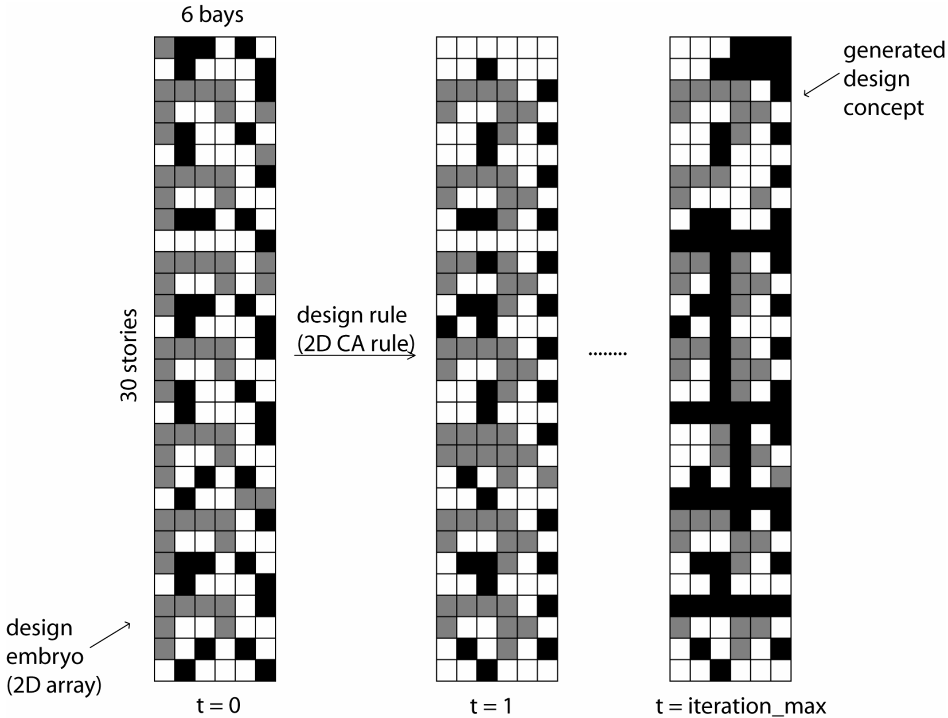


Figure 31. Process of generation of a design concept of a wind bracing system from a single 2D design embryo and a single design rule based on a 2D CA

Figure 32a) shows a schematic view of the structure of the genome that encodes the design embryo (gray squares) and the design rule represented by a 2D CA rule (white squares). Similar to the generative representations described in the previous sections, one has to assume an ordering of individual decision rules making up the design rule (in this case represented by a 2D CA rule) in order to uniquely define it and encode it in the genome. Figure 32c) shows an example of the ordering (denoted by r1-r19683) of all possible combinations of cell state values in the given local neighborhoods (here two-dimensional 3 by 3 square neighborhoods consisting of 9 cells, i.e. Moore neighborhoods).

The design embryo in this representation is a two-dimensional array encoding the entire initial configuration of a wind bracing system in a tall building (see Figure 32d)). The initial part of the genome encodes a *linearized* version of this array and its length is equal to the total number of cells in the configuration of a wind bracing system in a tall building, i.e., number of bays * number of stories. In the particular example of a tall building with 6 bays and 30 stories shown in Figure 32d), the initial configuration consists of 180 cells, and hence 180 leftmost genes in the genome (genes denoted e1-e180 in Figure 32b)) encode the design embryo, as it is shown in Figure 32b).

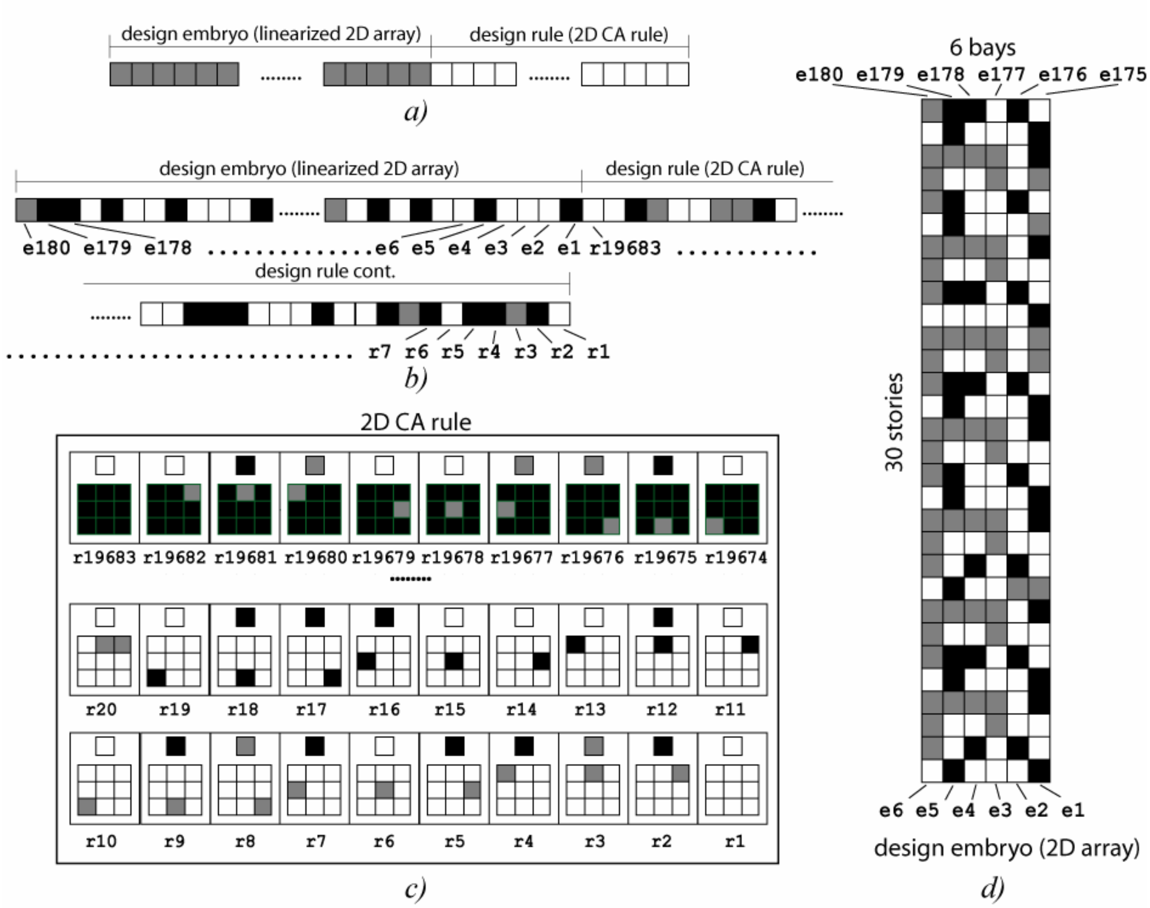


Figure 32. a) Schematic view of the structure of a genome encoding the generative representation of a wind bracing system and consisting of a single design embryo (linearized 2D array) and a single design rule based on a 2D CA, b) specific instance of the genome encoding the initial configuration shown in part d) (genes e1-e180) and the design rule (a 2D CA rule) shown in part c) (genes r1 - r19683)

The design rule consists of a complete set of decision rules whose conditions (bottom rows representing 3 by 3 squares of cells in Figure 32c)) incorporate all possible combinations of cell state values in the given local neighborhoods (here Moore neighborhoods) and the outcomes (cells placed above the corresponding squares in Figure 32c) specify the values of the central cells in these neighborhoods at a next time step. As discussed above, all possible combinations of conditions for the design rule shown in Figure 32c) are ordered from r1 to r19683. If this ordering is assumed fixed for the entire class of the design rules, then the outcome values uniquely define every rule belonging to this class. This fact has been used in previous sections to define encodings of the design rules represented by 1D CA rules. It has been generalized here to uniquely define design rules represented by 2D CA rules. The actual encoding of the design rule shown in Figure 32c) in the genome is presented in Figure 32b). Here, genes r1-r19683 encode the corresponding outcome values produced by the design rule presented in Figure 32c) and, given the assumed ordering, uniquely define it.

The number of genes necessary to encode the design rule depends on several parameters. As it was the case with the design rules represented by 1D CA rules, the number of possible cell state values and the size of the local neighborhood significantly affect the length of the encoding of a design rule. Besides, the length is also linked to the shape of the local neighborhood. In the case of the design rule shown in Figure 32c) with 3 cell state values, the neighborhood radius equal to 1, and Moore neighborhood, the number of genes necessary to encode this rule in the genome is equal to 19,683 genes. Increasing the number of cell state values to 7 causes a rapid growth of the length of the genome. For example, 40,353,787 genes are necessary compared to 19,863 genes when 3 cell state values were used. In such cases, the only feasible approach involves design rules represented by totalistic 2D CAs. Then, the genome length is equal to 199 and 235 genes, when 3 and 7 cell state values are used, respectively.

One of the major advantages of this representation is the fact that it allows for an explicit representation of two-dimensional interactions among design elements. It is possible to investigate various ranges of interaction among elements by selecting different shapes of the local neighborhood as well as by changing its radius. This property might be particularly important in modeling complex engineering systems where local and highly nonlinear interactions among structural members are impossible to describe using traditional mathematical formulas.

Major disadvantage of this approach is the problem of scalability. As discussed earlier, increasing the number of cell state values or the size of the local neighborhood causes a rapid growth in complexity. Hence, the design rules represented by totalistic 2D CA rules will be used in majority of the design experiments reported in this dissertation (see sections 6.4 and 8.2.3).

4.4.4. Multiple 1D Embryos and 1D CA Rules Representing Steel Structures

Previously described representations of steel structural systems in tall buildings were focused only on one, albeit important, part of the system, i.e., a system of wind bracings. A complete design concept of a steel structural system, however, should represent not only the system of wind bracings, but also beams, columns, and supports.

An approach to encode complete design concepts of steel structures in tall buildings is presented in this section. It makes use of an idea of combining several generative representations of various subsystems of a steel structure into one genome. In order to achieve it, an approach similar to the one described in section 4.4.1 is employed. Figure 33 shows a schematic view of the structure of the genome representing a complete design concept of an entire structural system.

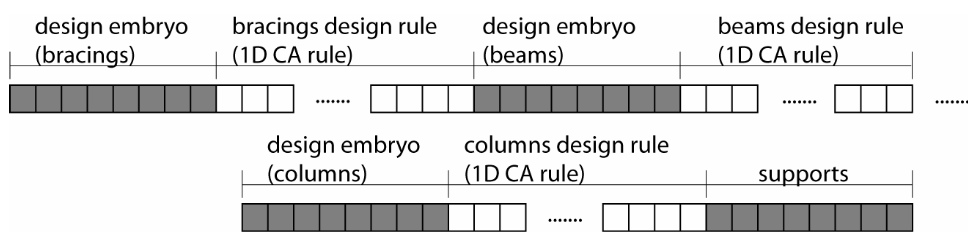


Figure 33. Schematic view of the structure of a genome encoding the generative representation of an entire steel structural system in a tall building and consisting of multiple design embryos and multiple design rules represented by 1D CAs

The linear genome encodes design embryos of a wind bracing system, a beam system, and a column system (gray cells) and design rules represented by 1D CA rules (white cells). The design rules generate the systems of wind bracings, beams, and columns from the corresponding design embryos. Additionally, a configuration of supports is also encoded at the end of the linear genome (gray cells) but it is not iterated because there is no need to develop a two-dimensional structure of supports (building supports are completely defined by a one-dimensional configuration of support types). Significant differences of this representation compared to the representations discussed earlier include the fact that it not only encodes the entire steel structural system but it also consists of non-homogenous genes. Various parts of the genome encode different subsystems of the steel structure and hence different attributes are used to represent them. These attributes, in general, can have different number of possible values, e.g., attributes representing bracing elements can have up to seven values while the attributes representing beams and columns can have up to five values. The attributes representing supports in a steel structure can have up to 4 values (see section 4.2).

The process of development of a complete design concept from its generative representation is presented in Figure 34.

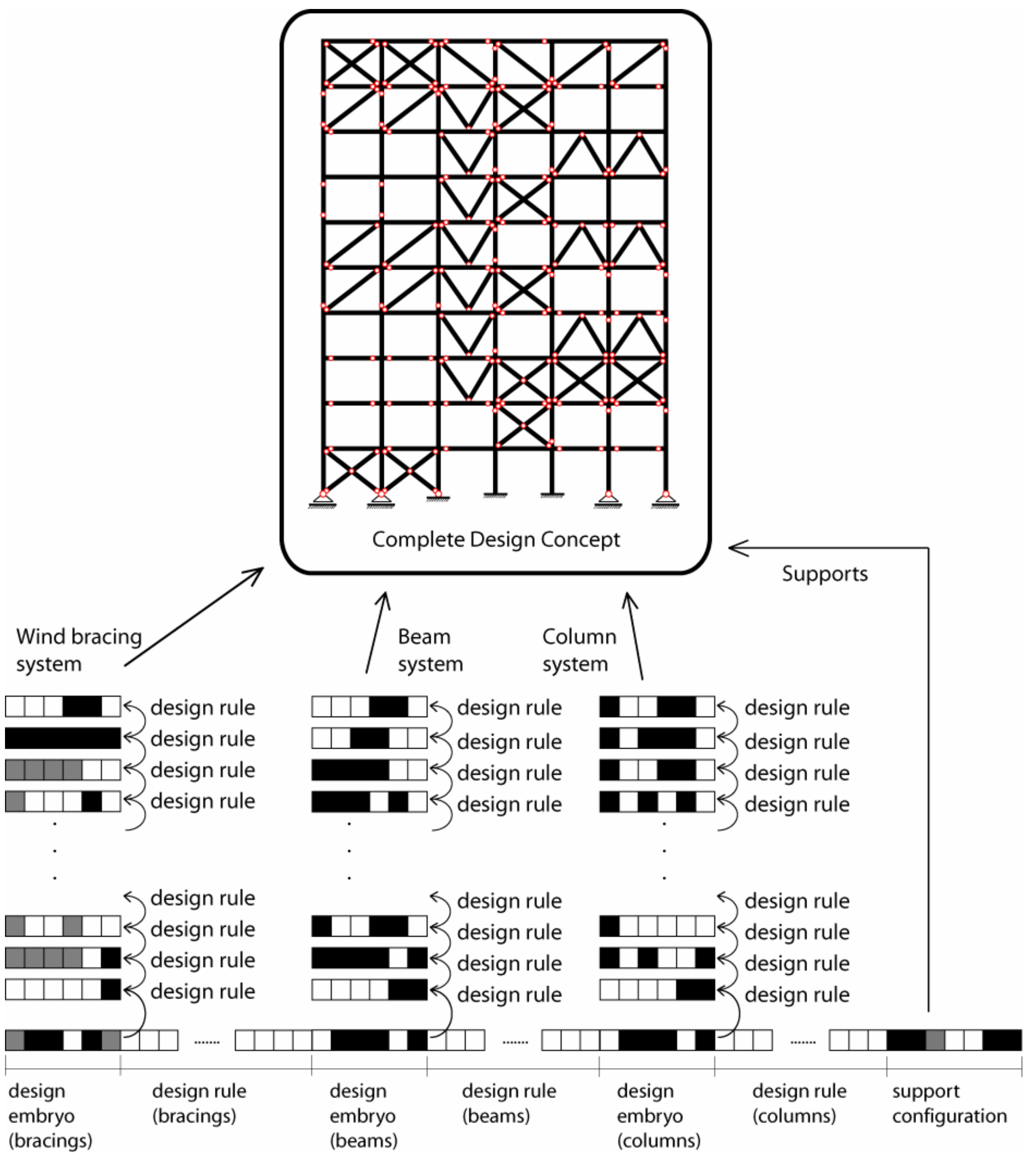


Figure 34. Process of developing the entire steel structural system in a tall building from a genome consisting of multiple design embryos and multiple design rules represented by one-dimensional cellular automata

Each design rule is applied to its corresponding design embryo and iterated the number of times that is one less than the number of stories in a tall building. In this way, the systems of wind bracings, beams, and columns are formed. The configuration of supports represented by the rightmost genes is not iterated. Once the complete configurations of all subsystems are developed, they are assembled together and form a complete representation of a design concept. At this point, the complete design concept can be evaluated.

The advantages of this approach are similar to the ones described in section 4.4.1, i.e. compactness and excellent scalability. A genome encoding a complete design concept of a tall building with 30 stories and 6 bays and consisting of a wind bracing system with 3 types of bracings, a beam system with 2 types of beams, a column system with 2 types of columns, and with 2 types of supports has 69 genes. When all possible types of structural elements are considered in the representation, i.e. 7 types of wind bracing elements, 5 types of beam elements, 5 types of column elements, and 4 types of supports, then the length of the genome is equal to 683 genes. In the case when totalistic 1D CA rules are used, the length of the genome is reduced to 81 genes compared to 683 genes required for standard 1D CA rules.

The disadvantage of this representation is also similar to the one described in section 4.4.1, namely the lack of diversification of design rules. Additional drawback involves the necessity to create a specialized mutation operator, even though the modifications required in adapting a standard mutation operator to this representation should be minimal.

4.5. Summary

In the first section of this chapter, I provided a general overview of representations of structural systems and discussed what distinguishes them from traditional models of structural systems used in engineering science. I also argued that representations consisting of attributes with symbolic values are suitable for the conceptual design problems. Thus, they could be used for the two design problems investigated in this dissertation.

The second section of this chapter introduced state-of-the-art representations of steel structural systems in conceptual design. For these types of representations, called parameterized representations, each gene represents an attribute corresponding to a dimension of the search space. I also described in detail the attributes representing major elements of steel structural systems in tall buildings, including bracings, beams, columns, and supports.

In the third section, I proposed a new approach to represent engineering systems based on models of complex systems and inspired by the processes of morphogenesis occurring in nature. These new representations do not encode complete design concepts, as in the parameterized representations, but rather rules on how to develop, or grow, these designs. At the end of this section, I also provided a definition of morphogenic evolutionary design.

The fourth section of this chapter extended the ideas presented in the preceding section and proposed several types of design concept generation mechanisms based on one-dimensional and two-dimensional cellular automata. It also discussed their computational and representational advantages and disadvantages. Furthermore, I demonstrated how each type of a design concept generation mechanism can be encoded in a generative representation consisting of two parts: a design embryo and a design rule. The design embryo defines an initial configuration of structural elements and the design rule defines a transformation that changes the current configuration of structural members into a new configuration. A complete design concept of an engineering system is developed by applying the design rule to the corresponding design embryo.

In the next chapter, I will introduce Emergent Designer, an integrated research and design support tool, which implements the design method proposed in this dissertation as well as the representations of steel structural systems discussed in this chapter.

5. EMERGENT DESIGNER

"Our minds are finite, and yet even in these circumstances of finitude we are surrounded by possibilities that are infinite, and the purpose of human life is to grasp as much as we can out of the infinitude."

(Alfred North Whitehead)

In this chapter, I introduce Emergent Designer, an integrated research and design support tool which implements Emergent Engineering Design. The system was used to conduct all design experiments reported in this dissertation (see chapters 6-8). The chapter also discusses the flow of information within EED and within the individual phases which constitute the proposed design method. It describes the results of the second stage of the Theoretical Structural Validation (see section 3.6.1) whose objective was to build confidence in the internal consistency of EED. The discussion of the information flow within EED and within its individual phases is instantiated by detailed descriptions of the flow of information within Emergent Designer and its components

Emergent Designer is a unique research and design support tool which applies models of complex systems to represent engineering systems and design processes, and to analyze their results. A high-level overview of the system and its architecture is provided in section 5.1. Section 5.2 provides diagrams of the flow of information within EED/Emergent Designer as well as the input/output relationships among the individual phases of the design method/components of the system. Finally, section 5.3 describes the actual implementation of Emergent Designer. Figure 35 shows organization of chapter 5 as well as major components of Emergent Designer.

5.1. General Overview

Emergent Designer is an integrated research and design support tool that implements Emergent Engineering Design, the design method proposed in this dissertation. It is based on the ideas proposed in this dissertation, including the ideas on how to represent engineering systems and design processes using various models of complex systems. As discussed in section 9.2, the system forms one of the major contributions of this dissertation.

Emergent Designer's architecture, discussed in section 5.1.1, has been built upon the structure of the proposed design method. Consequently, major components/modules of the system implement major phases of EED described in section 3.4. Further, the components/modules of the system implement models, procedures, and algorithms directly related to the research hypotheses posed in this dissertation and presented in section 3.3. Therefore, they are directly linked to the fundamental hypothesis of this dissertation.

Emergent Designer is intended for conducting design experiments in the area of structural design and for analysis of their results using methods, models, and tools from statistics and time series analysis. Thus, it can be used as a design support tool equipped with state-of-the-art mechanisms for the generation of novel design concepts and for conducting their optimization. It is at the same time a versatile research tool that implements advanced methods and tools for the analysis of the design processes and of the obtained experimental results.

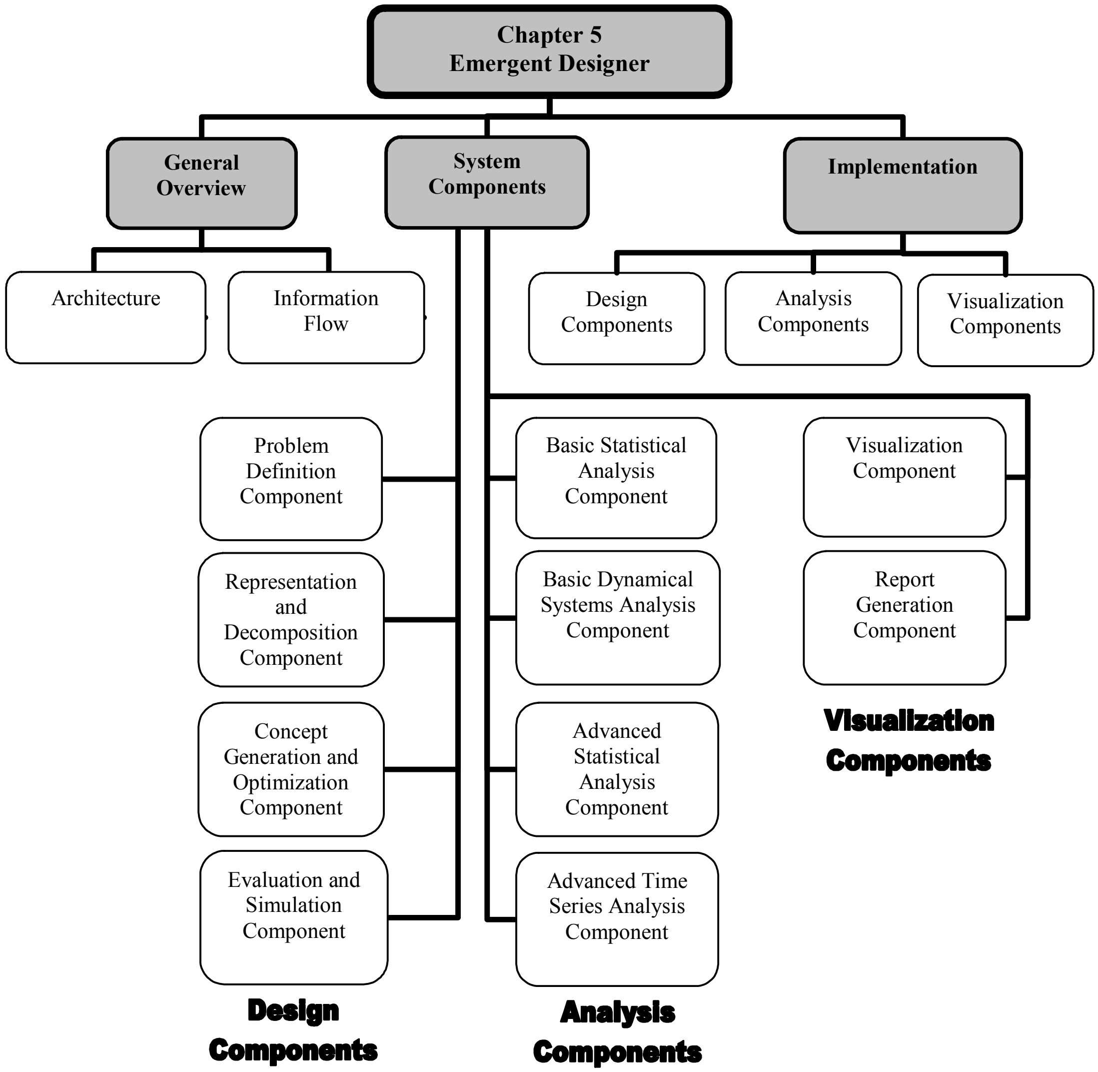


Figure 35. Organization of chapter 5

The following subsection provides a high-level overview of the system's architecture and briefly describes its major components/modules. Subsection 5.1.2 presents the flow of information within the system and discusses integration of its components and their interactions.

5.1.1. Architecture

Emergent Designer consists of 10 major components/modules which can be divided into three major groups:

- **Design components**
  They implement Emergent Engineering Design, the design method proposed in this dissertation. They form the core of the system and conduct the actual design processes.
- **Analysis components**
  They implement tools and methods for the analysis of the experimental results and design processes. The components included in this group are aimed to provide quantitative information about the conducted design processes as well as statistical estimates of the performance of the design method. They are also intended to provide

deeper understanding of the dynamics of design processes and the structure of the design spaces from a global/holistic perspective.

- **Visualization components**
  These components implement various visualization methods and report generation mechanisms. They include tools which support visualization of the results of various analyses, e.g. statistical or time series, conducted by the system's components. Also, automated tools for the generation of experimental reports that include detailed descriptions of experimental parameters and obtained results are implemented.

A high-level overview of the architecture of Emergent Designer is presented in Figure 36. It shows the individual components of the system contained in each of the groups discussed above. They will be discussed in more detail in section 5.2.

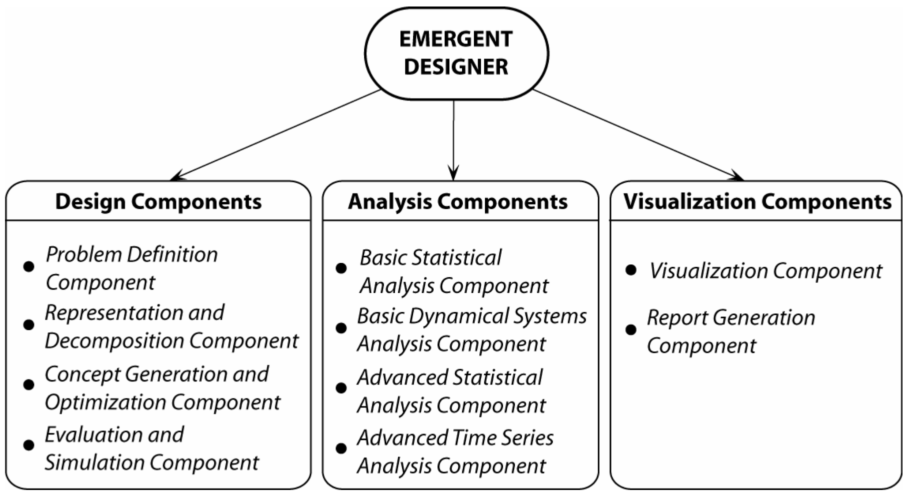


Figure 36. Architecture of Emergent Designer

5.1.2. Information Flow

The flow of information in Emergent Designer is presented in Figure 37. It provides an overview of the relationships among the components discussed in the previous section and shows where user input/decisions are expected. It doesn't show, however, the information flow within the individual components which are discussed in section 5.2.

Once Emergent Designer has been started, a user has a choice of conducting a new design experiment or using advanced statistical and time series analysis tools to analyze experimental data saved from previous experiments. By default, a new design experiment is selected and the *Problem Definition Component* is called to define a design problem.

Problem Definition Component is intended to select a domain of interest, e.g. steel skeleton structures in tall buildings, and a specific design problem that will be solved, e.g. design of a wind bracing system. This component allows for specification of values of the parameters defining the considered design problem, e.g. the number of stories in a tall building, or the height of a story. *Problem Definition Component* also implements mechanisms for saving the system's parameters and their values to a file, and retrieving previously saved values from a file.

When the design problem is completely defined, a user has to decide whether, or not, to decompose the problem into several sub-problems. *Representation and Decomposition Component* is used for this purpose. If the design problem is to be decomposed, then a user selects one of the several decomposed representations. On the other hand, if the design problem is considered as a whole, then one the representations of the entire engineering systems can be chosen. In this case, the spectrum of possible representations includes parameterized representations and generative representations (see section 5.2.2). *Representation and Decomposition Component* also supports the specification of values of representation specific parameters, e.g. the shape of the local neighborhood in generative representations based on cellular automata.

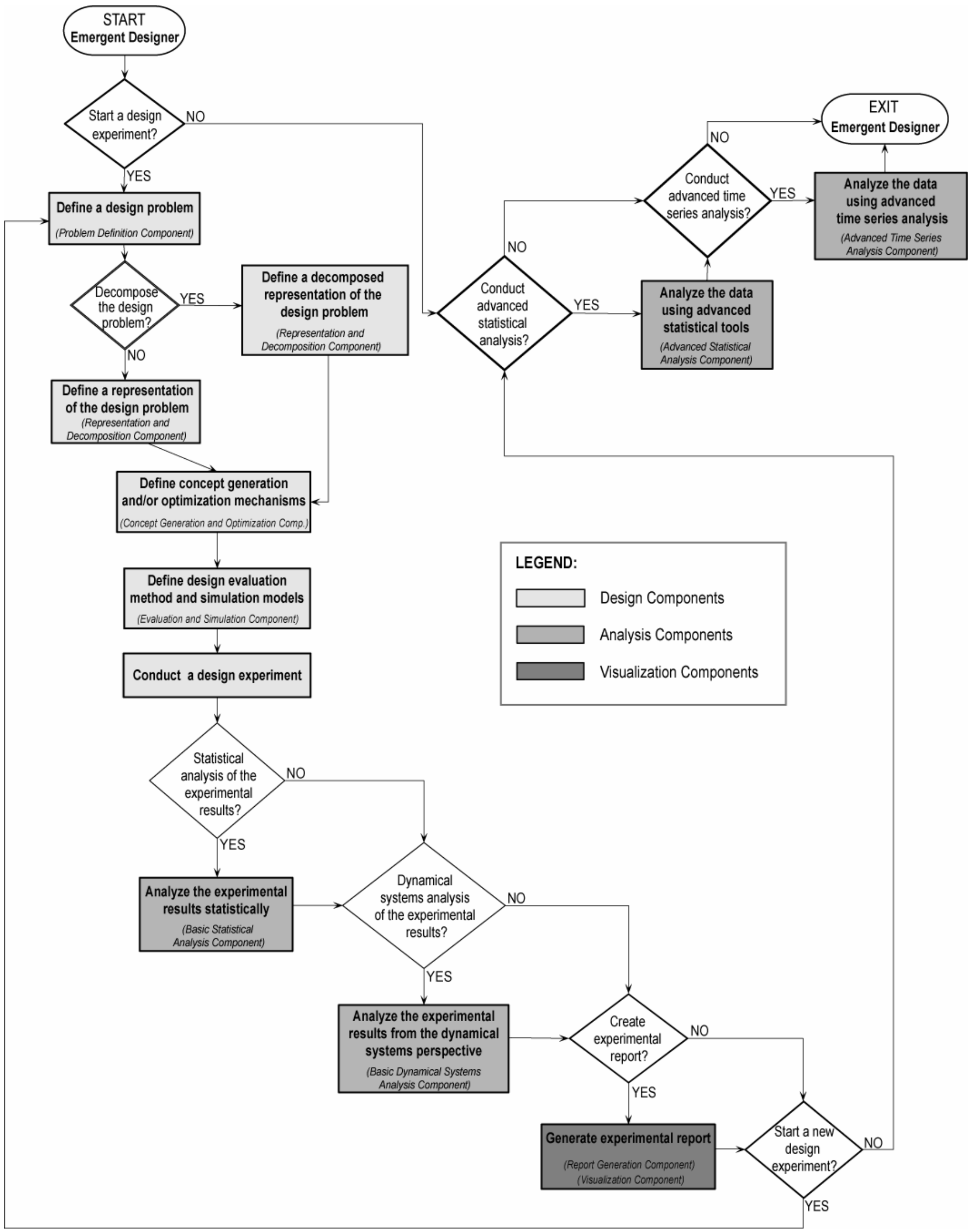


Figure 37. Information flow in Emergent Designer

When the design problem and its representation have been fully defined, the *Concept Generation and Optimization Component* is used to specify the type of a concept generation mechanism and to determine whether, or not, the topology optimization and/or sizing optimization should be conducted. If only a concept generation mechanism is selected, i.e. no optimization is performed, then the design concepts are produced by the design concept generators based on generative representations, e.g. iteration of 1D or 2D cellular automata (see chapter 6). On the other hand, when the optimization of engineering systems is to be performed then a user has two possible choices:

1. If the focus is on design optimization issues then only an optimization mechanism, e.g. an evolutionary algorithm, is used together with traditional parameterized representations of engineering systems (see chapter 7).
2. If both generation of novel design concepts and their subsequent optimization are considered as important objectives then an optimization mechanism is combined with generative representations (see chapter 8).

The design concepts produced by the design concept generation and/or optimization mechanisms are transferred to the *Evaluation and Simulation Component* which evaluates them and assigns fitness value(s) (multiple fitness measures are used in the multiobjective evaluation). This component is used to select an evaluation model assumed in a given design experiment and the values of evaluation specific parameters, e.g. methods for the determination of wind loads acting on structural system, or magnitudes of dead and live loads, etc. Also, simulation parameters, including the number of runs, the termination criteria, etc., need to be defined in order to run a design experiment.

The four components described earlier, i.e. Problem Definition Component, Representation and Decomposition Component, Concept Generation and Optimization Component, and Evaluation and Simulation Component, form a group of design components that implements the actual design method.

Once the values of all the parameters implemented in this group of components have been determined (default values are also used where possible), the actual design experiment can be initiated. *Basic Statistical Analysis Component* and *Basic Dynamical Systems Analysis Component* support online monitoring of design processes by providing best-so-far fitness values and trajectories of points (design concepts) in the design spaces. *Basic Statistical Analysis Component* also provides the mechanisms for collecting relevant experimental data and saving them in files.

When a design experiment is finished, *Basic Statistical Analysis Component* can be used to calculate and display average best-so-far fitness values with corresponding 95% confidence intervals. At that point, a user can also generate a complete experimental report listing all the parameters and their values used in the design experiment as well as its results. *Report Generation Component* and *Visualization Component* are employed during the process of the automatic generation of an experimental report. *Report Generation Component* gathers the names and values of the parameters used in the experiment and extracts relevant experimental results. It also collects important statistical data calculated by the *Basic Statistical Analysis Component*. *Visualization Component* can be used to produce a landscape visualization graph, if applicable, and charts representing progress of individual runs in the design experiments. When all the textual, numerical, and graphical data are available, *Report Generation Component* compiles them together into a single document that is subsequently displayed as an experimental report.

At this point, the user can choose to start a new design experiment, or to analyze the experimental data using advanced statistical and time series analysis tools, or simply exit the system. If a new design experiment is selected, *Problem Definition Component* is called again and the entire process described above is repeated. On the other hand, if advanced statistical analysis, or advanced time series analysis, is chosen then *Advanced Statistical Analysis Component* or *Advanced Time Series Analysis Component* is utilized, respectively.

5.2. System Components

This section describes in detail each of the system's components that were briefly introduced in the previous section. It discusses the information flow within each of the system's components and describes the parameters and their allowable values set by each component. Section 5.3 discusses the actual implementation of the components.

5.2.1. Problem Definition Component

Problem Definition Component implements the preliminary phase of the proposed design method in which a design problem is defined. The output of this component, i.e. a complete description of a design problem in terms of parameters and their values, becomes the input to the *Representation and Decomposition Component*. This component provides necessary domain knowledge and specifies parameters of the considered design problem. It defines the input data which are transferred to the components implementing the actual design method. It is used to perform the following tasks:

- Domain selection, e.g. steel skeleton structures in tall buildings.
- Problem selection, e.g. design of a wind bracing system, or design of the entire steel structural system.
- Specification of the problem parameters, e.g. the number of stories, or story height.

Figure 38 shows the flow of information within the *Problem Definition Component*. The external input to the component defines the overall purpose (what to design), requirements, and constraints (feasibility criteria) the design should satisfy. Based on this input, a design domain is selected. If it is necessary to define domain specific parameters and their values, then they are defined in the next step. For example, when a domain of steel skeleton structures has been selected, it is necessary to specify the following parameters and their values:

- *Dimensionality*: 2D, 3D
- *Design type*: truss, frame, other
- *Structural analysis type*: analysis, optimization, verification
- *Behavior type:* *first-order, P-Δ*
- *Sidesway:* *prevented, permitted*
- *Cross-section database*: AISC, CISC, other
- *Unit system*: metric, U.S. customary
- *Length unit*: mm, m, in, ft
- *Force unit*: N, kN, lbs, kips

As discussed in more detail in section 5.3, *Problem Definition Component* assumes some default values of the parameters listed above but gives the user flexibility to adjust them appropriately.

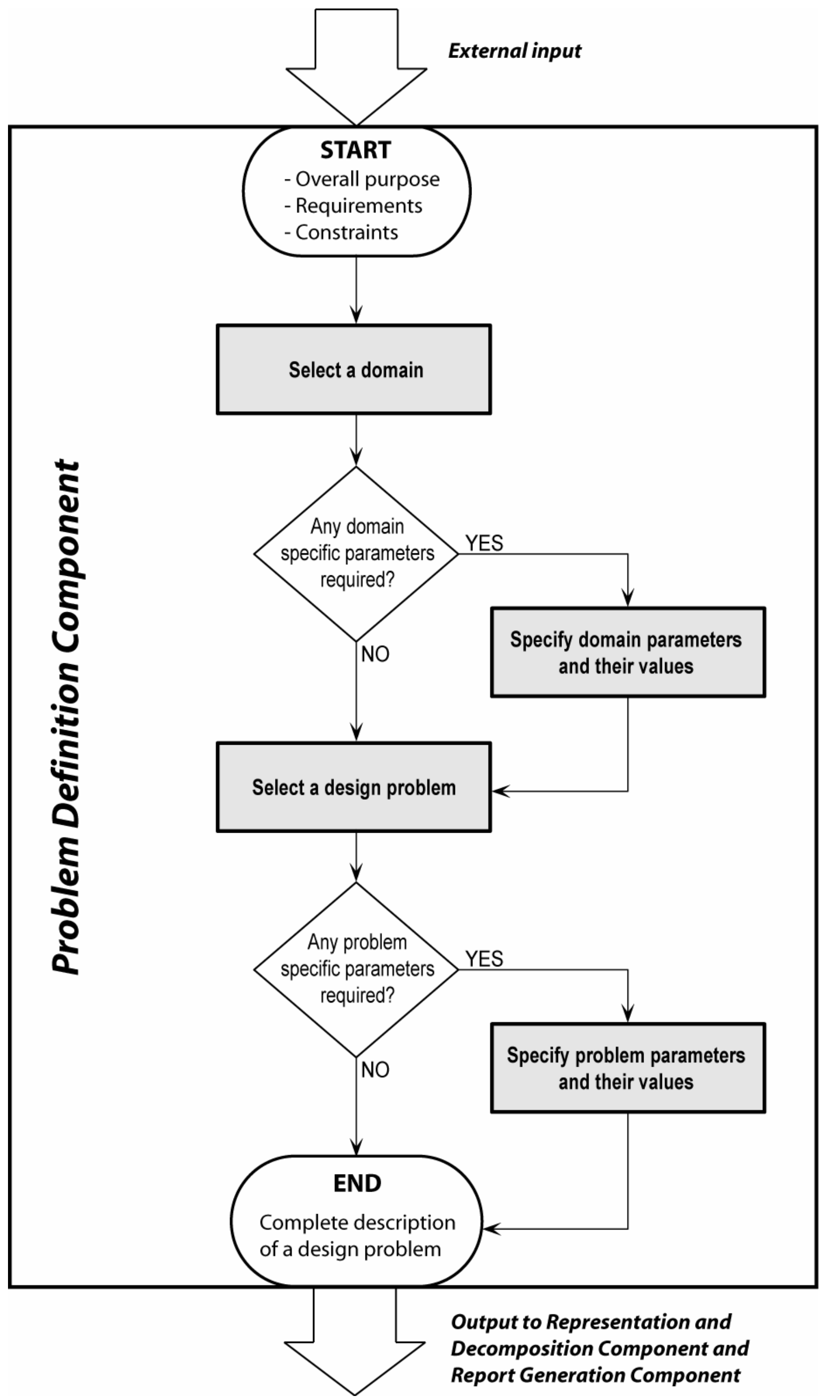


Figure 38. Information flow within the Problem Definition Component

When the domain and its parameters have been defined, a design problem can be selected, e.g. conceptual design of steel structural systems in tall buildings. Here again, some problem specific parameters and their values have to be defined. They include:

- *Number of stories:* 30 (default)
- *Number of bays:* 7 (default)
- *Story height:* 14.0 in (default)
- *Bay width:* 20.0 in (default)

- *Types of bracing elements*: a subset of the bracing types shown in Figure 19
- *Types of beam elements:* a subset of the beam types shown in Figure 20
- *Types of column elements:* a subset of the column types shown in Figure 21
- *Types of supports:* a subset of the support types shown in Figure 22

The values of the first 4 parameters are default values assumed by the system. They can be arbitrarily changed and can take on any feasible value in the context of the problem domain. When all the parameters defining a domain and design problem have been defined, a complete definition of a design problem has been established. It forms the output of the *Problem Definition Component* which subsequently becomes an input to the *Representation and Decomposition Component* and *Report Generation Component*.

5.2.2. Representation and Decomposition Component

Representation and Decomposition Component is used to conduct the first and second phases of EED, i.e. Representation Space Definition and Representation Space Decomposition (see Figure 15), in which representation of an engineering system and its decomposition, if any, are defined. The input to this component consists of a complete definition of the design problem which is obtained from the *Problem Definition Component*. The output produced by the *Representation and Decomposition Component* defines the representation of the engineering system being designed. This component is used to conduct the following tasks:

- Selection of a representation for the design problem, e.g. a parameterized representation, or a generative representation based on one-dimensional or two-dimensional cellular automata.
- Selection of a decomposition of a given problem.
- Specification of parameters for a given type of representation (e.g., resolution for binary representations, or the neighborhood shape and the neighborhood radius for generative representations).

The flow of information within the *Representation and Decomposition Component* is shown in Figure 39. The complete description of the design problem is a starting point of the development of a representation of the engineering system being designed. When the design problem is complex, it might be decomposed into sub-problems. In this case, the decomposition of the problem has to be specified.

Decomposition specific parameters and their values, if any, are defined in the next step. Examples of decomposition specific parameters for the problem of designing steel structures in tall buildings include:

- *Number of sub-problems:* 2, 3, 4
- *Types of elements in sub-problems:* [(bracings), (beams, columns, supports)], [(bracings), (beams, columns), (supports)], [(bracings), (beams), (columns), (supports)]

When all the decomposition parameters have been specified, or when there is no decomposition, the actual encoding of the engineering system has to be defined. Here, several types of encodings can be used as discussed in chapter 4. The encodings supported by the *Representation and Decomposition Component* can be divided into two major groups: parameterized and generative (see section 2.1.3).

When the encoding type has been determined, it is usually necessary to define some additional parameters and their values. For example, when generative representations based on one-dimensional cellular automata (see section 4.4.1) have been selected, the following encoding specific parameters must be defined:

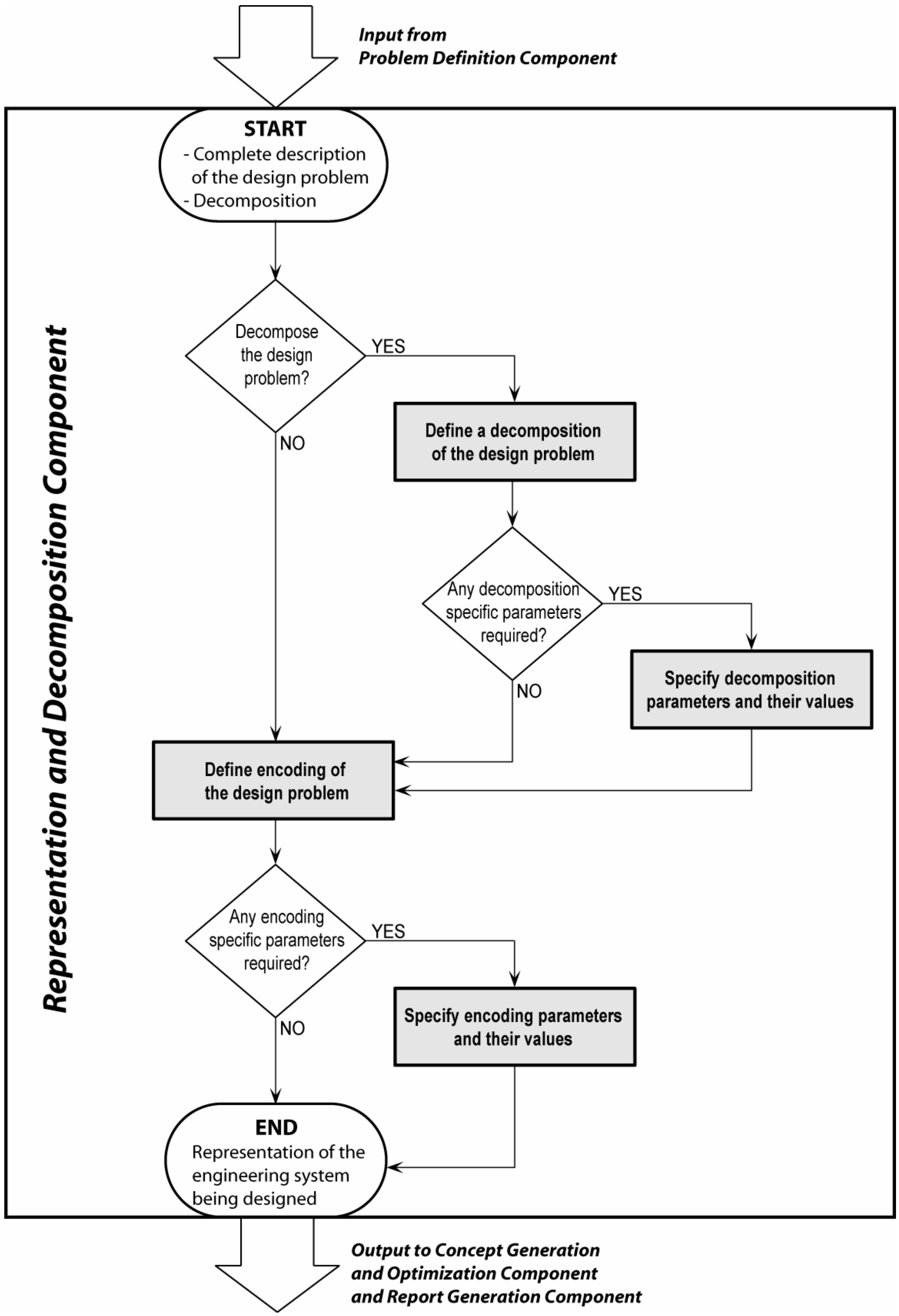


Figure 39. Information flow within the Representation and Decomposition Component

- *CA type :* regular, totalistic
- *Local neighborhood radius*: 1 (default)

- *Number of cell state values*: determined by the problem definition

When all the decomposition and encoding parameters have been defined, the representation of an engineering system is completely specified. The representation becomes the output of the *Representation and Decomposition Component* which is subsequently utilized by the *Concept Generation and Optimization Component* and *Report Generation Component*.

5.2.3. Concept Generation and Optimization Component

Concept Generation and Optimization Component is utilized to conduct the third phase of the proposed design method, namely Generation and Optimization of Design Concepts (see Figure 15). This component defines representations of engineering design processes. As discussed in chapter 3, EED assumes the model of the design process based on generate-and-test, or trial-and-error, principle.

The following tasks are handled using this component:

- Selection of the mechanisms for generation of design concepts, e.g. various types of cellular automata (1D, totalistic 1D, 2D, totalistic 2D).
- Selection of the mechanisms for optimization of design concepts, e.g. various types of evolutionary algorithms.
- Specification of parameters of optimization mechanisms, i.e. parent and offspring population sizes, types of genetic operators, etc.

Representation of an engineering system obtained from the *Representation and Decomposition Component* forms the input to *Concept Generation and Optimization Component*. The produced output consists of feasible design concepts with assigned fitness value(s).

The internal structure of this component is much more complex than of the other components in this group. The flow of information within the *Concept Generation and Optimization Component* is shown in Figure 40. It consists of three major subcomponents: *Concept Generation Component*, *Topology/Shape Optimization Component*, and *Sizing Optimization Component*. Depending on the type of a representation of an engineering system provided as input and the decisions made regarding the optimization mechanisms, either only one subcomponent, or two, and even all three subcomponents, can be utilized in the design process.

If merely the *Concept Generation Component* is used, then the design concepts are produced solely by the concept generation mechanisms, e.g. 1D or 2D cellular automata. In this case, no optimization mechanisms are employed during the design process. Generated design concepts are evaluated, given some evaluation criteria, and the best designs are identified at the end of a design process. Thus, in this case the focus of the design processes is shifted towards novelty. Design concept generation mechanisms are studied experimentally in chapter 6.

On the other hand, if an engineering system is represented using a parameterized encoding then no concept generation mechanism is necessary to produce the design concept from its representations (there is a direct mapping between the representation and the attributes describing an engineering system). In this case, the design processes focus exclusively on optimality issues. Design optimization mechanisms can be applied at the topology/shape level (conceptual/embodiment design) using the *Topology/Shape Optimization Component* and/or the member sizing level (detailed design) utilizing the *Sizing Optimization Component*. These mechanisms are investigated in chapter 7.

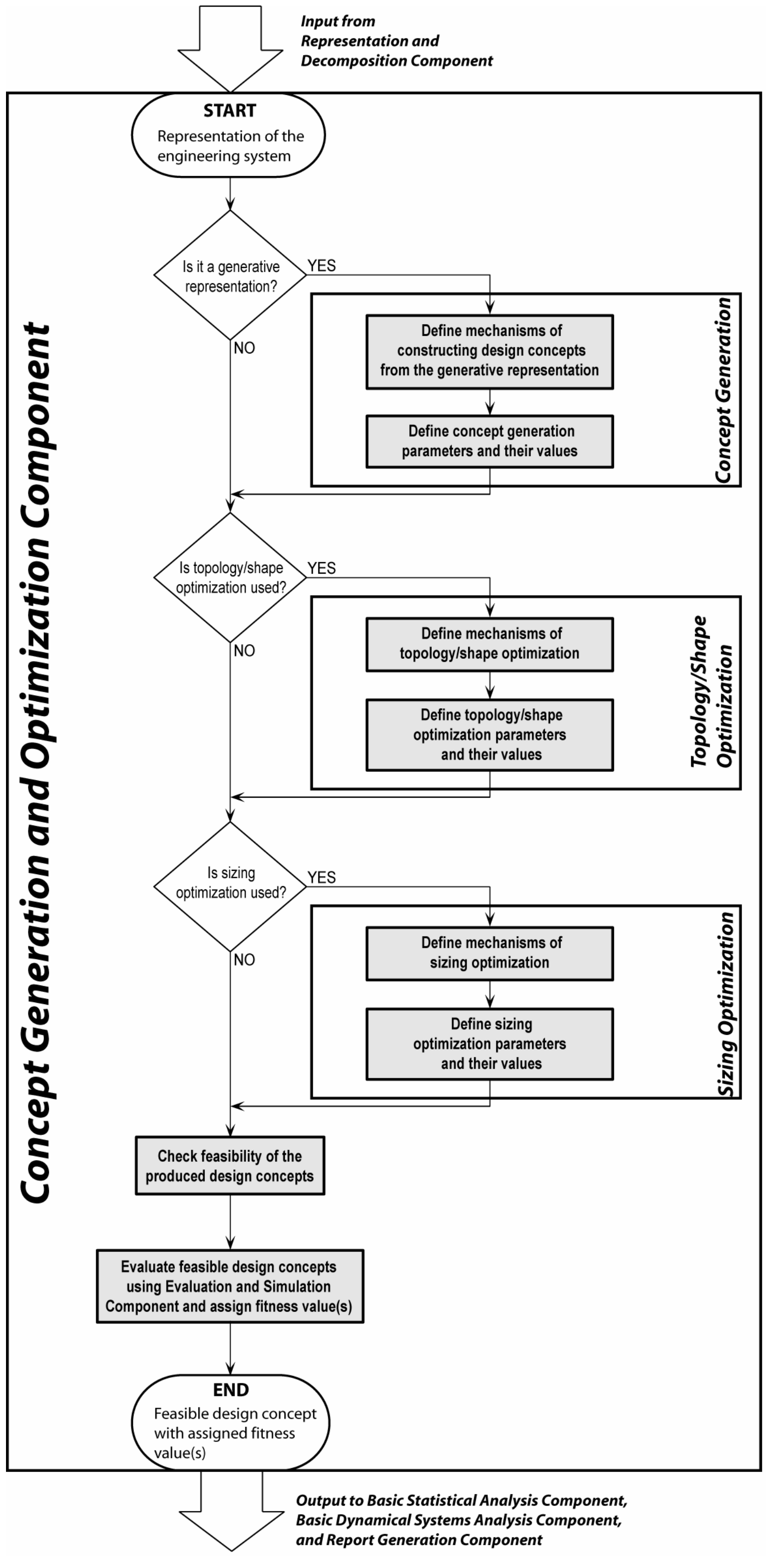


Figure 40. Information flow within the Concept Generation and Optimization Component

It is also possible to combine design concept generation mechanisms with design optimization mechanisms, which is one of the key ideas presented in this dissertation. This corresponds to the situation in which novelty of generated design concepts and their optimality are equally important design objectives. To achieve both objectives, the concept generation mechanisms need to be specified using the *Concept Generation Component* and optimization mechanisms must be determined using the *Topology/Shape Optimization Component* and/or the *Sizing Optimization Component*. The combined mechanisms utilizing generative representations, named morphogenic evolutionary design (see section 4.3), are investigated in chapter 8.

If the representation of an engineering system, obtained from the *Representation and Decomposition Component*, is generative, then the mechanisms of producing design concepts from this representation have to be defined using the *Concept Generation Component*. For example, when 1D or 2D CA representations are used, then cellular automata need to be determined to develop design concepts from their representations. The parameters required here include the parameters used in the process of defining the representation of an engineering system which is obtained as input. Additionally, the values of two more parameters, namely the *Design rule* and the *Design embryo*, must be defined. They determine a specific CA rule and initial configuration of cells used to generate design concepts.

The following parameters are necessary to fully define cellular automata:

- *CA dimension:* (1D or 2D) determined by the representation of an engineering system; higher-dimensional CA can also be used for some problem domains but they are not studied in this dissertation
- *CA type :* (regular or totalistic) determined by the representation of an engineering system
- *Local neighborhood shape:* (Moore, von Neumann, etc.) determined by the representation of an engineering system (2D CA only)
- *Local neighborhood radius:* determined by the representation of an engineering system
- *Number of cell state values*: determined by the representation of an engineering system
- *Design rule (CA rule)*: randomly generated, arbitrarily assumed
- *Design embryo (initial configuration)*: randomly generated, arbitrarily assumed

Some types of representations of engineering systems require additional parameters to fully determine the concept generation mechanism. An example of such a parameter is the maximum number of iterations of a cellular automaton (*iteration_max*), which was defined in sections 4.4.2 and 4.4.3.

When the combined approach is used (generative representation and optimization mechanisms), or when parameterized representations of engineering systems are employed, then the parameters defining optimization mechanisms need to be specified. For example, when an evolutionary algorithm is employed to optimize topology/shape of an engineering system, the following parameters and their values must be defined by the *Topology/Shape Optimization Component*:

- *Type of evolutionary algorithm:* GA, ES, EP, or unified EA
- *Generational model:* overlapping, non-overlapping
- *Parent population size:* 5 (default)
- *Offspring population size:* 25 (default)

- *Parent selection mechanism:* uniform deterministic, uniform stochastic, fitness proportional, binary tournament
- *Survival selection mechanism:* uniform stochastic, fitness proportional, binary tournament, truncation
- *Mutation type:* none, bitflip, Gaussian, delta, integer-based
- *Mutation frequency:* one gene, all genes
- *Mutation step size:* 1.0 (default)
- *Mutation adaptation value:* 0.1 (default)
- *Crossover type:* none, one-point, two-point, uniform
- *Crossover frequency (uniform only):* 0.2 (default)

Sizing optimization, if used, solves the problem of finding the optimal cross-sections of all structural members of an engineering system being designed. This type of structural optimization is conducted when the topology/shape of a given design concept has been already determined, or when it is assumed constant. Sizing optimization has been traditionally, and successfully, performed using mathematical optimization methods (see section 2.1.7). It is also possible to employ evolutionary algorithms to solve this task and, in fact, one of the earliest applications of EA considered the sizing optimization of simple truss systems (see the survey in section 2.1.7). When sizing optimization is conducted, the optimization mechanism needs to be specified first by the *Sizing Optimization Component*. Once it is defined, additional sizing optimization parameters and their values, if any, have to be determined. When an evolutionary algorithm is used to solve the sizing optimization problem, the parameters and their values are similar to the ones used for topology/shape optimization. On the other hand, when a traditional mathematical optimization method is employed, then parameters specific to this method must be defined.

When all the parameters defining the design concept generation, topology/shape optimization, and sizing optimization are set, the feasibility check conditions and methods of handling constraints (see section 2.1.4) need to be specified. They define mechanisms of determining feasibility of produced design concepts and mechanisms that are applied when an infeasible design concept is identified. The constraint-handling method can be defined using the following parameters and their values:

- *Constraint-handling method type:* penalty function, repair algorithm, etc. (see section 2.1.4)

Depending on the type of a selected constraint-handling method, some additional parameters might be necessary. For example, when a penalty function method is used, then additional parameter values might be required, including:

- *Penalty function type:* static, dynamic, death penalty, etc. (see section 2.1.4)
- *Value of the penalty factor:* constant, determined by a formula (penalty term), etc.

Each generated design concept is tested for feasibility. When it satisfies all feasibility criteria defined by the *Problem Definition Component,* then it is passed to the *Evaluation and Simulation Component* where it is evaluated and assigned fitness value(s). On the other hand, when a produced design concept is proved infeasible, then constraint-handling methods need to be employed, e.g. a repair algorithm or a penalty method. In the case when a repair algorithm is used, an attempt is made to repair the design concept and, if successful, the design concept is passed to the *Evaluation and Simulation Component* and assigned a fitness value(s). If the repair is unsuccessful though, the design concept is determined infeasible and assigned worst possible fitness value(s) (death penalty method).

The output of the *Concept Generation and Optimization Component* consists of feasible design concepts with assigned fitness values which are subsequently passed to the *Basic Statistical Analysis Component*, *Basic Dynamical System Analysis Component*, and *Report Generation Component*.

5.2.4. Evaluation and Simulation Component

Evaluation and Simulation Component implements the last phase of EED, namely Fitness Evaluation (see Figure 15). It defines design evaluation models and general mechanisms of managing and monitoring simulations of design processes. The input to this component consists of a phenotypic representation of a design concept which is obtained from the *Concept Generation and Optimization Component*. The output produced by the *Evaluation and Simulation Component* consists of the same design concept provided as input but this time with an assigned fitness value, or fitness values in the case of multiobjective evaluation. The following tasks are conducted using this component:

- Specification of the load conditions considered during an evaluation process, e.g. wind loads acting on a steel structure in a tall building.
- Selection of an evaluation model and mechanisms used to measure goodness/fitness of generated design concepts, e.g. a structural analysis package to calculate the total weight of the structural system.
- Specification of general simulation parameters, e.g. the number of runs, lengths of individual runs, etc., and monitoring of the simulation progress.

The flow of information within the *Evaluation and Simulation Component* is shown in Figure 41. First, an evaluation model needs to be selected. The model defines the number of objectives and the number and type of evaluation criteria that will be used to determine goodness of a design concept provided as input. The evaluation model is defined by considering the following parameters:

- *Number of objectives:* 1, 2, 3, etc.
- *Number of evaluation criteria:* 1, 2, 3, etc.
- *Type of evaluation criteria:* total weight, maximal deflection, total cost, etc.
- *Type of weights:* uniform, non-uniform, summing to 1, etc.

The value of the parameter specifying the number of objectives determines whether a standard single-objective evaluation should be conducted (when the value is equal to 1), or rather multiobjective evaluation methods ought to be employed (when the value is greater than one). The parameters specifying the number and type of evaluation criteria define the quantities that are measured and subsequently utilized by the evaluation model. It is worth mentioning that multiple evaluation criteria may be used even when a single-objective evaluation is performed. In this case, the evaluation criteria are combined into a single objective function through the normalization and arbitrarily assigned weights.

The next step defines a loading model which is applied to the structural system being designed. Several parameters describe types of loads considered in a given design situation as well as the number and types of load combinations. They include the following parameters and their values:

- *Types of loads:* dead, live, wind, etc.
- *Number of load combinations:* 1, 2, 3, etc.
- *Types of load combinations:* (wind + live + dead), etc.

When the types of loads considered in a given design situation are set, then the magnitudes and locations of specific loads need to be determined. Also, specific loading model parameters have to be defined, including coefficients applied to various load types for a given load combination, etc.

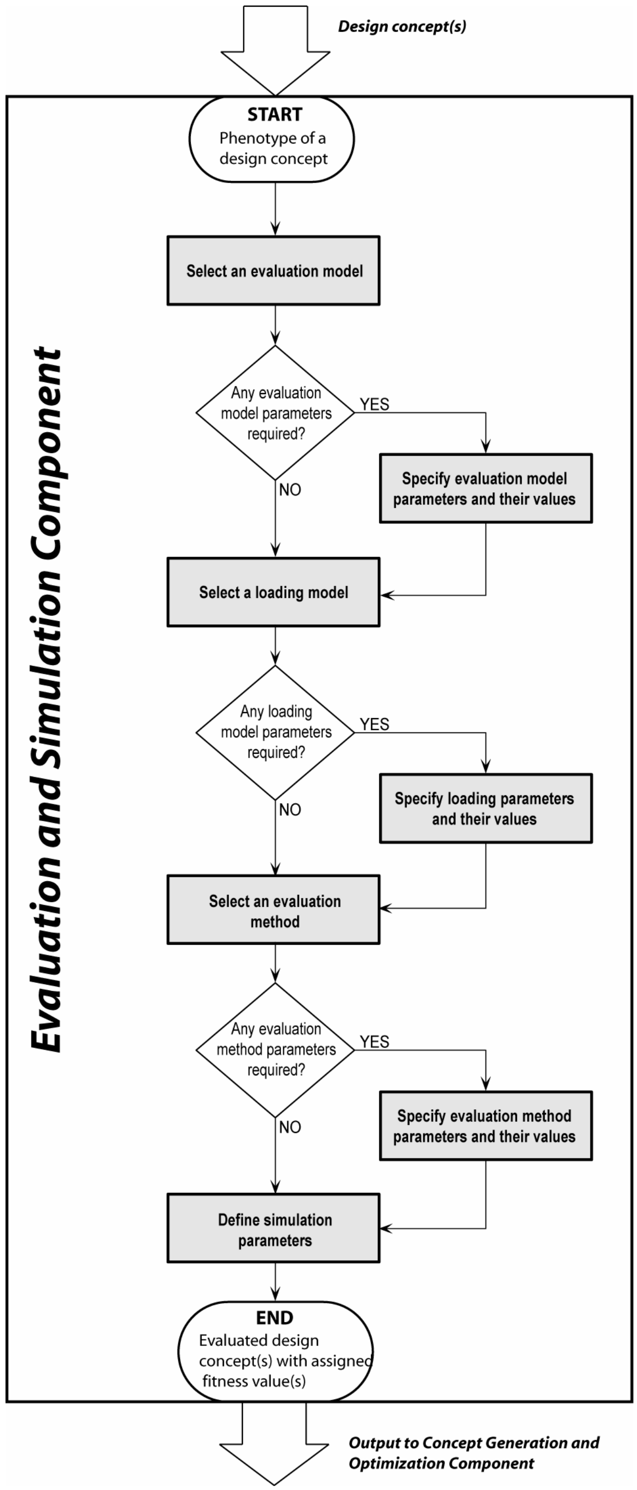


Figure 41. Information flow within the Evaluation and Simulation Component

When the evaluation and loading models are established, the evaluation method must be specified. Here, several types of structural analysis methods can be considered, including the displacement method, finite elements method, etc. If the selected evaluation method requires additional parameters, they are defined at a subsequent step.

Finally, general simulation settings need to be determined. They describe the overall length of a design experiment in terms of the number of runs (evolutionary based design processes are stochastic in nature and any inferences based on the experimental results have to be justified statistically). They also specify initialization and termination criteria for individual runs in terms of random seed values used during the initialization process and the maximum number of fitness evaluations, respectively. The following parameters need to be set:

- *Number of runs:* 10 (default)
- *Number of fitness evaluations:* 1000 (default)
- *Random seed values used* (name of the file storing the seeds)
- *Number of CA iterations:* 30 (default)

When all evaluation and simulation parameters are specified, then all phases of the proposed design method are completely defined. The system is ready to conduct a design experiment as it is shown in Figure 37.

The remaining components of Emergent Designer, described below, implement methods, models, and tools for the analysis of experimental results, their visualization and automatic report generation.

5.2.5. Basic Statistical Analysis Component

Basic Statistical Analysis Component implements basic statistical tools for the analysis of the results of design processes. The input to this component is obtained from the *Concept Generation and Optimization Component* and consists of design concepts with their fitness values as well as their data regarding when they were generated during the simulation (their "birth dates"). The following tasks are performed using this component:

- Collection of the experimental data and calculation of the best-so-far fitness statistics.
- Calculation of various statistical estimates that quantitatively describe design processes, including average best-so-far fitness and confidence intervals around the mean.
- Comparison of statistical estimates (means and confidence intervals) calculated from the results obtained in design experiments with multiple runs.

The first two tasks described above are performed online, i.e. during the actual design processes while the last task is conducted offline, when no design experiments are running.

The flow of information within the *Basic Statistical Analysis Component* is shown in Figure 42. When a new design concept has been generated and evaluated, its fitness value(s) and birth date are collected. These data are subsequently saved in the files storing the experimental results. Next, the data are analyzed and best-so-far statistics are calculated. They are also saved in the files storing statistical analysis results. At the same time, best-so-far statistics are passed to the *Visualization Component*.

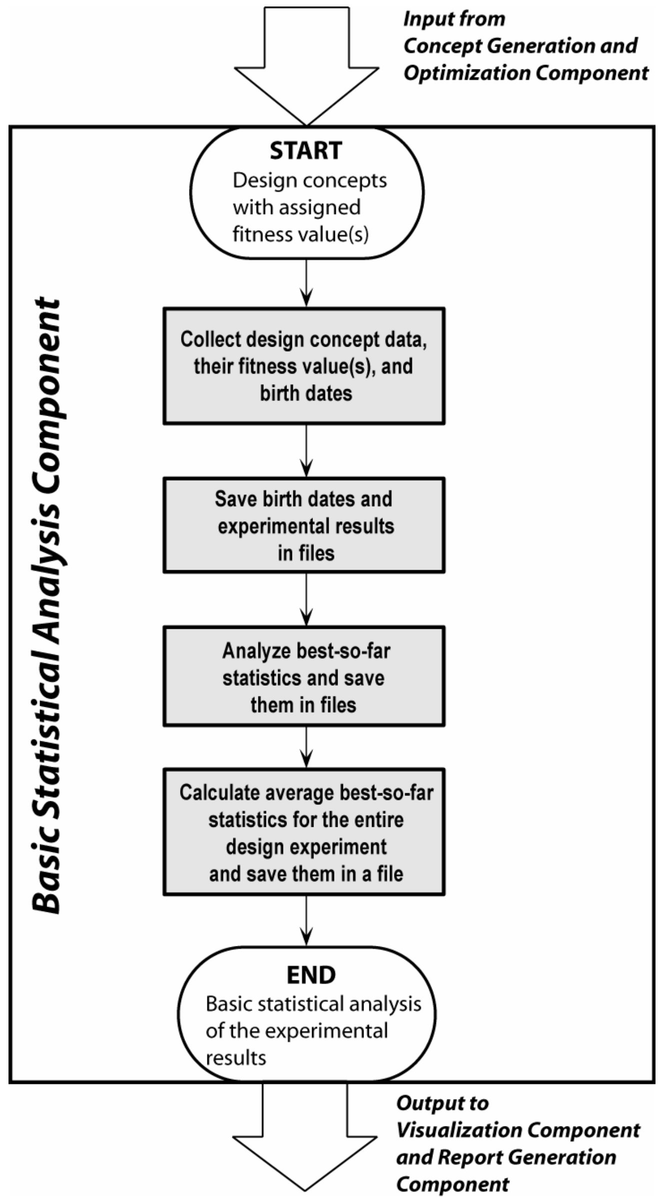


Figure 42. Information flow within the Basic Statistical Analysis Component

When the design process is finished, the average best-so-far statistics for the entire experiment are calculated and saved in a file. At the same time, they are also transferred to the *Visualization Component*. The output produced by the *Basic Statistical Analysis Component* consists of basic statistical analysis results, which are subsequently passed to the *Visualization Component* and the *Report Generation Component*.

5.2.6. Basic Dynamical Systems Analysis Component

Basic Dynamical Systems Analysis Component implements basic tools and methods for the analysis of the results of the design processes from the dynamical systems perspective. In this type of analysis, the design processes are considered as dynamical systems operating in the design spaces. Its input consists of design concepts, their fitness values, and their birth dates and is obtained from the *Concept Generation and Optimization Component*. The subjects of analyses are the properties of trajectories (coordinates of the generated points in the design space) of design processes and identification of attractors in the design spaces. The following tasks are conducted using this component:

- Collection of the trajectories data (coordinates of the generated points in the design space).
- Reconstruction of attractors in the design spaces from the experimental data.

These tasks are also performed online and show the actual dynamics of the design processes.

The flow of information within the *Basic Dynamical Systems Analysis Component* is shown in Figure 43. First, the experimental results data are collected. The trajectory information is extracted from the collected data and passed to the *Visualization Component*. Further, the trajectory data are analyzed and methods and tools of attractor reconstruction are employed, e.g. delay coordinates. The results of these analyses are subsequently transferred to the *Visualization Component*. The output produced by the *Basic Dynamical Systems Analysis Component* consists of basic dynamical systems analysis data and is utilized by the *Visualization Component*.

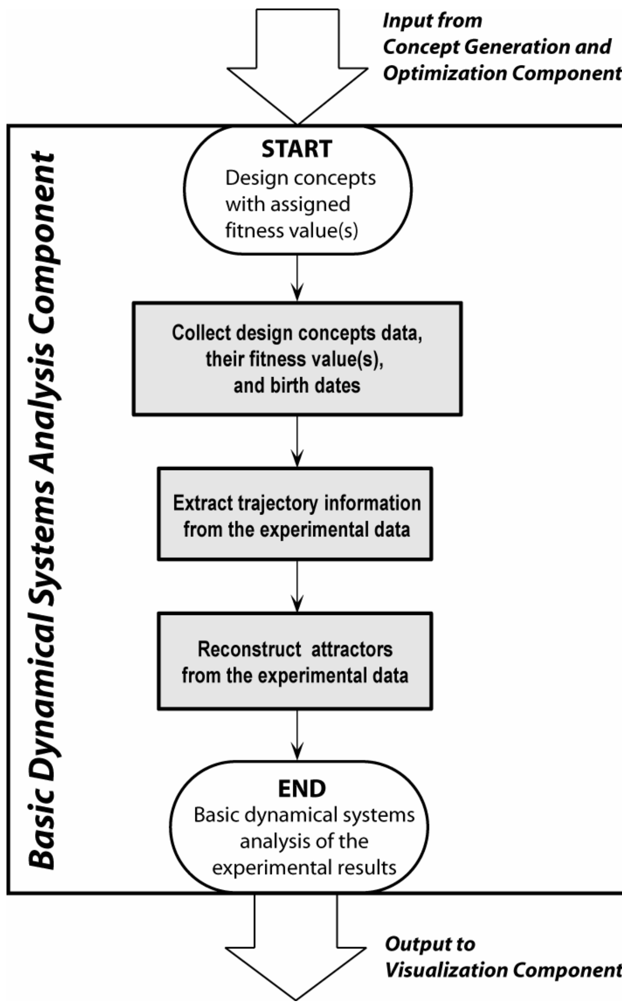


Figure 43. Information flow within the Basic Dynamical Systems Analysis Component

5.2.7. Advanced Statistical Analysis Component

Advanced Statistical Analysis Component implements advanced statistical analysis methods, models, and tools for the analysis of the experimental results. The statistical analysis conducted by this component is performed offline, i.e. after the design experiments have been conducted. The input is obtained from the files storing the experimental results which were previously saved using the *Basic Statistical Analysis Component*. *Advanced Statistical Analysis Component* contains the tools for the analysis of the sample distributions and making inferences about their means and medians. The following types of tasks are performed using this component:

- Reading the experimental data from file(s).
- Qualitative and quantitative analysis of the sample distributions, e.g. histograms, normal scores plots, skewness and kurtosis estimates, etc.
- Estimation of statistical quantities from the data (e.g. means and medians) using various point estimates and interval estimates.
- Saving the analyses results in files.

The flow of information within the *Advanced Statistical Analysis Component* is shown in Figure 44. First, the experimental results data are read from the files. Next, the qualitative and quantitative analysis of the sample distributions is conducted, if desired. This type of analysis involves various histograms, normal scores plots, and skewness and kurtosis estimates. It helps to determine the overall qualitative properties, e.g. the shape of a sample distribution.

When the shape of a sample distribution is better known, then appropriate methods and tools for estimation of various statistical quantities, i.e. means, medians, etc., can be employed. The results of these analyses are transferred to the *Visualization Component* and subsequently displayed in a form of charts, graphs, and histograms and/or saved in files.

5.2.8. Advanced Time Series Analysis Component

Advanced Time Series Analysis Component implements advanced tools and models from the linear and nonlinear time series analysis. The analysis, similar to the one performed by the *Advanced Statistical Analysis Component*, is conducted offline. Also, the input consists of the experimental results stored in previously saved files. The following types of tasks can be conducted using this component:

- Reading the time series data from a file(s).
- Qualitative and quantitative analysis of the time series data using various methods and tools, e.g. delay coordinates, power spectrum, autocorrelation, etc.
- Saving the analysis results in a file.

The flow of information within the *Advanced Time Series Analysis Component* is shown in Figure 45. First, the experimental data need to be read from files. Next, a method of the time series analysis must be selected. If the chosen analysis method requires some additional parameters, e.g. time lag and embedding dimension in the delay coordinates plot, they are specified at the next step. When all the parameters have been defined, the time series analysis of the experimental data can be conducted. The results of these analyses are subsequently transferred to the *Visualization Component* and/or saved in a file.

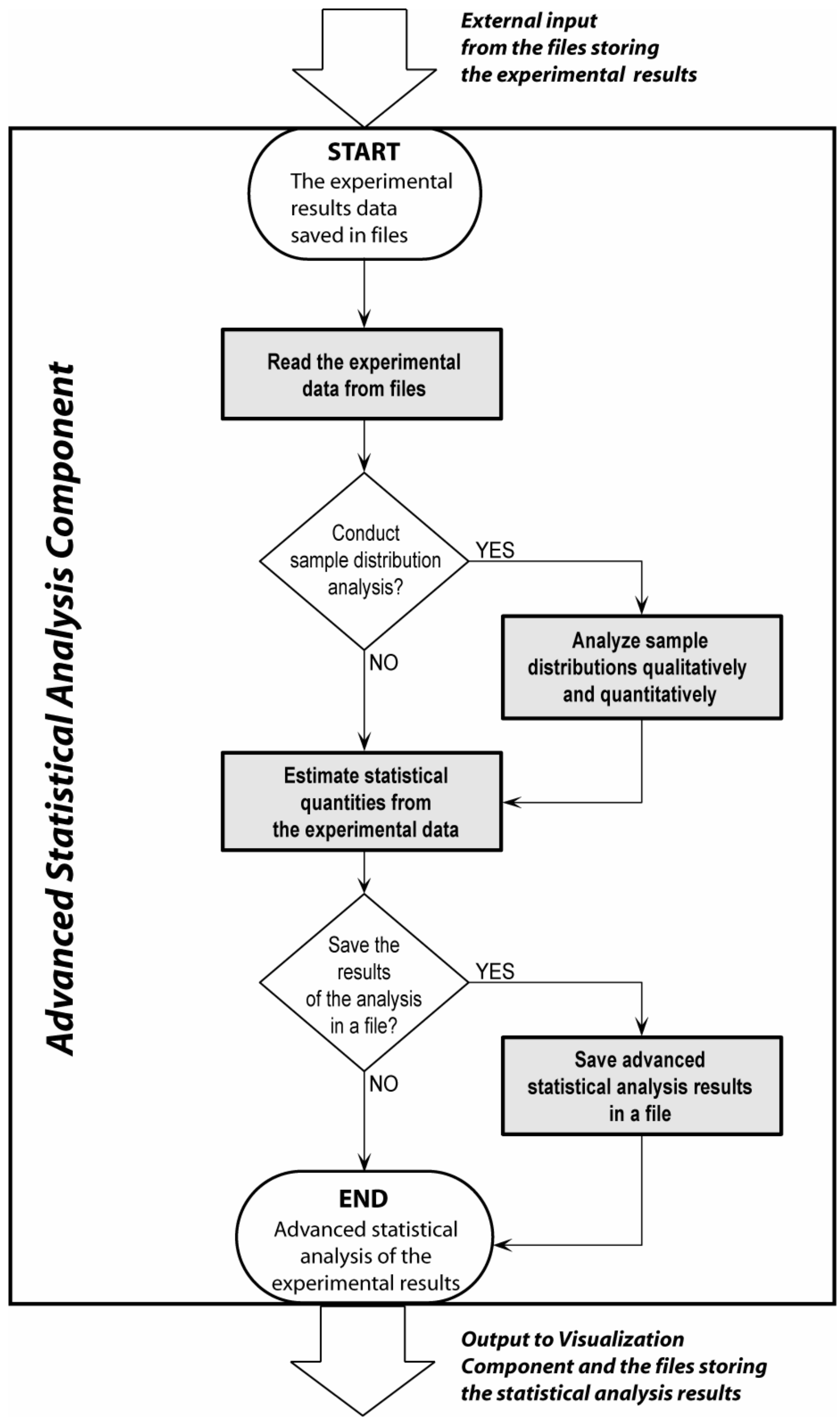


Figure 44. Information flow within the Advanced Statistical Analysis Component

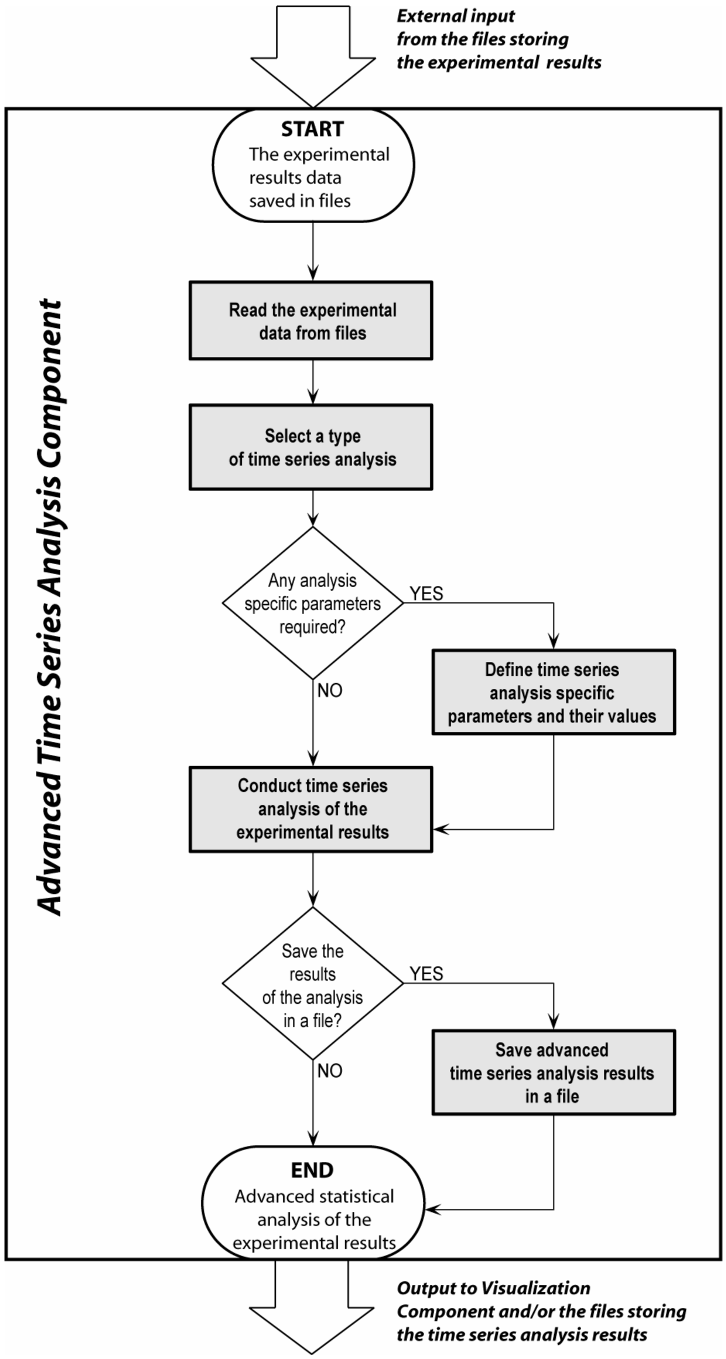


Figure 45. Information flow within the Advanced Time Series Analysis Component

5.2.9. Visualization Component

Visualization Component implements various methods of data visualization. It supports a qualitative analysis of the experimental results and offers necessary functionality to save produced graphs and charts in files and experimental reports. The input to this component consists of the experimental data and it is obtained from various components of the system. The following types of tasks can be conducted using this component:

- Display of generated design concepts.

- Interactive display of simple three-dimensional fitness landscapes.
- Display of statistical, dynamical, and time series analyses conducted using various components of the system.

The flow of information within the *Visualization Component* is presented in Figure 46. First, the experimental results obtained as input are collected and information relevant to display and visualization purposes is extracted from the data. Next, depending on the data source, appropriate graphs and charts are produced including line charts, scatter plots, histograms, and renderings. The produced graphs are displayed by Emergent Designer's graphical user interface (GUI). Each generated graph may also be saved in a file. This last option is implicitly used by the *Report Generation Component* which utilizes various graphs produced by the *Visualization Component* during the process of automatic generation of experimental reports. The graphs included in the reports are first saved in files and subsequently read by the *Report Generation Component*.

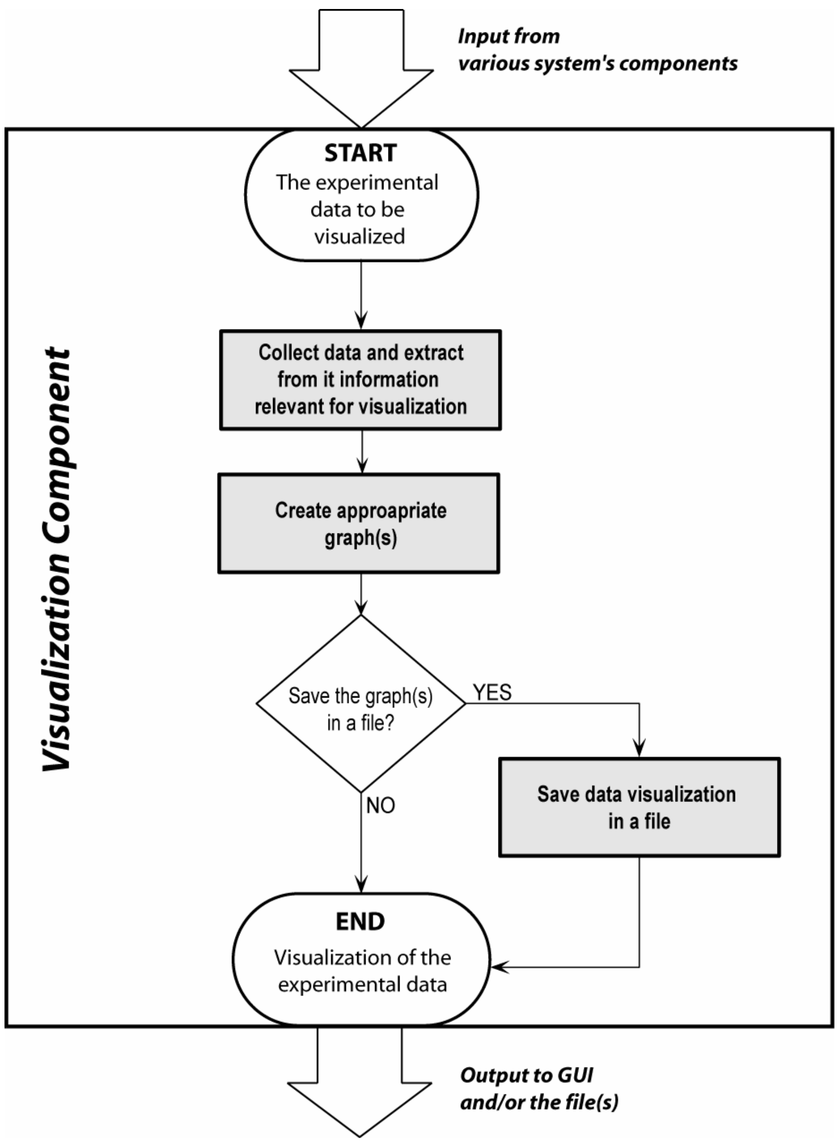


Figure 46. Information flow within the Visualization Component

5.2.10. Report Generation Component

Report Generation Component supports the automatic generation of experimental reports. It is intended to provide complete information about the experimental parameters and their values as well as the obtained results. The input to this component is obtained from various components of the system. The following types of tasks are conducted using this component:

- Collection of the experimental parameters and their values used in the experiment.
- Collection of the numerical results of various runs in a given experiment.
- Collection of statistical analysis data and various graphs illustrating progress of individual runs and average performance during the entire design experiment.
- Automatic generation of a full report containing all above mentioned elements.

The flow of information within the *Report Generation Component* is presented in Figure 47. First, the parameters and their values used in the experiment are collected from the components implementing the proposed design method, i.e., *Problem Definition Component*, *Representation and Decomposition Component*, *Concept Generation and Optimization Component*, and *Evaluation and Simulation Component*. They are grouped together and placed in the initial part of the experimental report. Next, *Basic Statistical Analysis Component* provides quantitative data on the results obtained in various runs as well as simple statistics, e.g. best-of-run fitness, etc. The quantitative data describing the individual runs are accompanied by the qualitative information received from the *Visualization Component* in the forms of graphs displaying the best, average, and worst fitness values obtained in individual runs. Thus, each run of a design experiment is described both quantitatively and qualitatively. This analysis concludes the section two of the experimental report. In the last section, a 'global' analysis of the entire design experiment is reported in which average best-so-far fitness values and the corresponding confidence intervals are given. Also, graphs showing best-so-far fitness of all runs as well as average fitness values are included. An experimental report containing all data mentioned above is automatically generated and may be subsequently saved in a file. Each generated graph may also be saved in a file. Hence, the output produced by the *Report Generation Component* consists of a complete experimental report which is displayed in the system's GUI and/or saved in a file.

5.3. Implementation

Emergent Designer has been implemented with a fully functional graphical user interface using Java. The decision to use this particular programming language was made due to the fact that several of the system's components were built upon existing packages written in Java. Moreover, *Emergent Designer* integrates several commercially available systems (e.g., *Mathematica© (Wolfram 2003)* and *OpenOffice.org*) and communicates with them using available Java APIs.

Another important aspect that influenced the choice of the programming language was the fact that Java is portable and network-oriented. Portability offers the flexibility of running the system on various platforms. Built-in networking capabilities open the possibility of using distributed architectures. Both of these issues are particularly important in structural design where the process of evaluation of generated design concepts is usually computationally expensive and conducted using specialized structural analysis software.

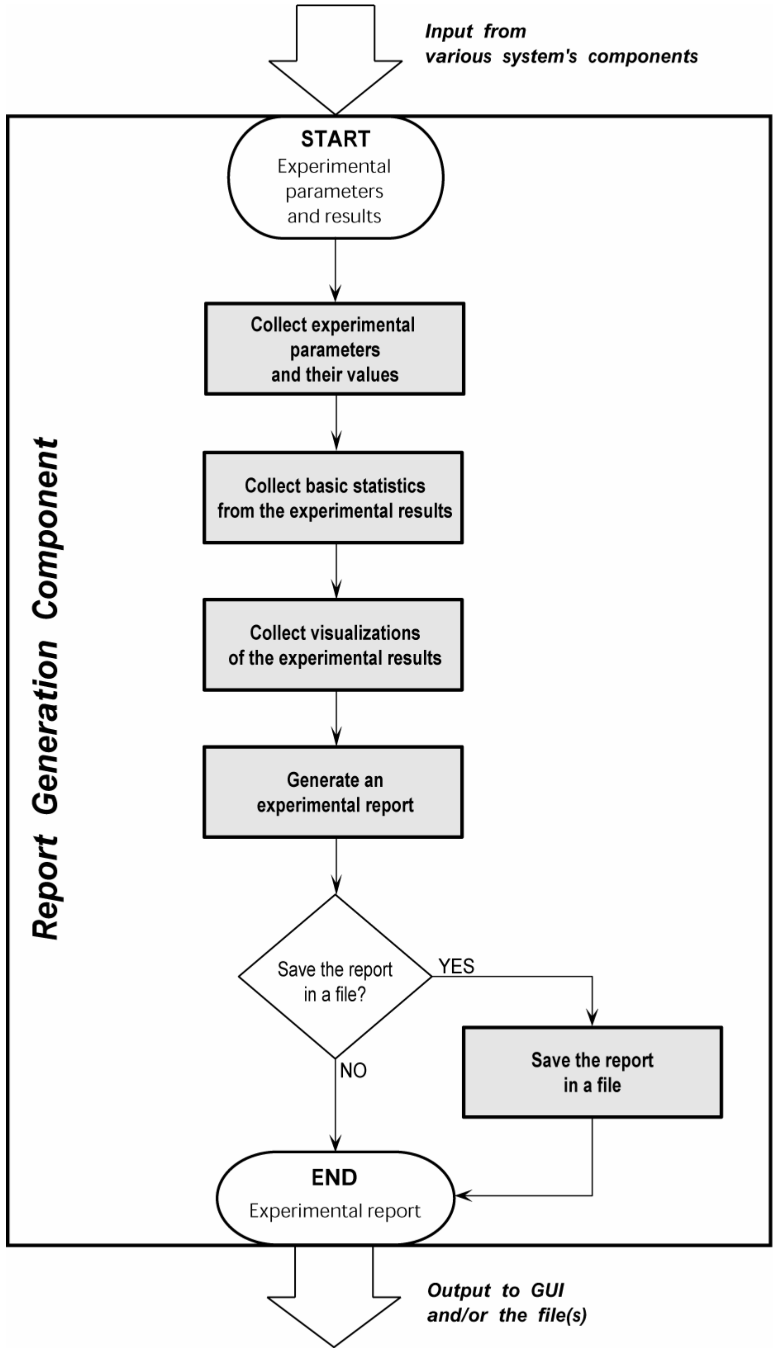


Figure 47. Information flow within the Report Generation Component

5.3.1. Design Components

Components implementing the proposed design method constitute the core of Emergent Designer. Their functionality, described in sections 5.2.1 - 5.2.4, was either directly implemented or borrowed from several existing packages and commercial systems that were integrated with Emergent Designer.

Two domains have been implemented in the *Problem Definition Component* the domain of steel structural systems in tall buildings and the domain of real-valued functions (added for testing purposes and analysis of the behavior of various components of the system). The domain of steel structural systems includes two major classes of design problems: design of a wind bracing system in a tall building and design of an entire steel structural system in a tall building.

Representation and Decomposition Component supports four types of representations:

- binary (parameterized)
- real-valued (parameterized)
- integer-valued (parameterized)
- cellular automata (generative)

The first two types are used mostly for real-valued problems while the latter two are applied to encode the designs concepts of steel structural systems in tall buildings. Real-valued and binary representation implementations were inherited from the existing evolutionary computation package called ec3 (De Jong to appear). On the other hand, integer-valued and cellular automata representations were directly implemented in the system.

Concept Generation and Optimization Component has been built upon four major existing packages and commercially available systems. Design concept generation utilizing various types of cellular automata is conducted by *Mathematica*© kernel which was integrated with Emergent Designer via *JLink*™. All major types of CA are supported, including 1D CA, totalistic 1D CA, 2D CA, and totalistic 2D CA.

Topology/shape optimization using evolutionary algorithms is supported by ec3 package (a Java-based evolutionary computation toolkit (De Jong to appear)). Here, all canonical evolutionary algorithms can be utilized, including genetic algorithms, evolutionary programming, and evolution strategies. The system also offers a possibility of employing a unified EA (De Jong to appear) in which all major elements of an EA, i.e. generational model, parent selection, offspring selection, population sizes, operators, etc., can be tuned to a particular design problem.

Sizing optimization, if applied, is conducted using a sophisticated optimization algorithm based on traditional mathematical programming method and implemented in SODA©. It is a commercially available structural analysis, design and optimization system developed by Waterloo Systems in Waterloo, Ontario, Canada. It was integrated with Emergent Designer to perform evaluation of designs and their sizing optimization. A detailed description of the optimization algorithm and some theoretical background can be found in (Grierson 1989).

Evaluation and Simulation Component implements evaluation models used to determine fitness of generated solutions. Current status of the system supports a single objective evaluation of individual design concepts only using one of the two evaluation criteria: the total weight (an estimate of the cost) or the maximal horizontal displacement (an estimate of the stiffness) of the steel structural system. The determination of a least-weight structure is performed by SODA and is conducted in conformance with the strength (stability) and stiffness (displacement) provisions of several commonly used steel codes, including AISC-ASD-89, AISC-LRFD-86, AISC-LRFD-93, CSA-Sl6.1-M89, or CSA-Sl6.1-94. Loading model required for evaluation of generated design concepts includes dead, live, and wind loads determined in conformance with the corresponding design codes. Wind forces are calculated for a given design case using a modified version of a commercial system *Wind Load*© V2.2.S developed by Novel CyberSpace Tools.

5.3.2. Analysis Components

Methods and models of basic statistical and dynamical systems analysis, described earlier in sections 5.2.5-5.2.6, have been implemented directly in Java. These analytical processes are conducted online, i.e. during the actual design processes. Basic statistical analysis involves best-so-far fitness statistics calculated for individual runs and average best-so-far fitness statistics and 95% confidence intervals computed for the entire design experiment. This analysis is also automatically saved in files.

Implemented methods of simple dynamical systems analysis include trajectory analysis which shows the dynamics of the processes in the design spaces as well as delay coordinates analysis. Delay coordinates are computed from the best-so-far fitness values with an arbitrarily assumed time lag.

Contrary to the basic analyses described above, advanced statistical and time series analyses are performed offline. Advanced statistical analysis includes estimation of sample distributions using histograms, normal scores plots, symmetry plots, and estimators of sample kurtosis and sample skewness. *Advanced Statistical Analysis Component* also implements various estimators of means and medians (e.g. the sample mean, the sample median and the trimmed mean) and the corresponding confidence intervals (normal approximation, Student's t test, Johnson's modified t test, and the sign test). Several advanced statistical analysis tools and methods have been implemented directly and but some of them were borrowed from *JMSL© Numerical Library* which was integrated with *Emergent Designer*.

Advanced Time Series Analysis Component implements the following methods of analysis of the experimental data: visual analysis of the time series data, delay coordinates plots with adjustable parameters (e.g. the embedding dimension and the time lag), power spectrum analysis, autocorrelation analysis with a flexible specification of autocorrelation lag and standard error bars according to either Barlett's or Moran's formula, and two types of recurrence plots (i.e. regular and thresholded) with a flexible specification of the embedding dimension, time lag, and the norm to calculate the distances between the points of a time series. As it was the case with the *Advanced Statistical Analysis Component*, several tools and methods of advanced time series analysis were directly implemented in the system while several of them have been borrowed from *JMSL© Numerical Library*.

5.3.3. Visualization Components

There are three major methods of visualizing experimental data in *Emergent Designer*. First, line plots and scatter plots (or more generally signal plots) are used to visualize experimental data transferred from the *Basic Statistical Analysis Component* and *Basic Dynamical Systems Analysis Component.* The plots are produced by a Java-based signal plotter called *PtPlot* developed at UC Berkeley. They are embedded in the *Emergent Designer*'s GUI and can be subsequently saved as bitmap files. Second, histograms are employed to visualize sample distributions. They are produced by the *Advanced Statistical Analysis Component*. These types of graphs are created using *JMSL© Graphical Library* integrated with *Emergent Designer*. They are also embedded in the system's GUI and provide functionality to save the produced graphs as bitmap files. Finally, interactive renderings of fitness landscapes can be produced for simple real-valued functions. These types of visualizations are produced using *Mathematica*'s advanced graphical capabilities and their display in the system's GUI is supported by *JLink*.

Automatic report generation capabilities, described in section 5.2.10, have been achieved through the integration of *Emergent Designer* with *OpenOffice.org*© and its Java API. *Report*

Generation Component collects and organizes textual, numerical and graphical data produced during the design processes and includes them in an experimental report. This report is subsequently displayed as an *OpenOffice.org* document. The report can be later saved in a file in any of the supported formats. In this way, it provides a complete summary of the parameters and the obtained results produced in the design experiments.

5.4. Summary

In this chapter, I conducted the second stage of the Theoretical Structural Validation of Emergent Engineering Design. By presenting and discussing information flow among the phases of the EED and within its individual components, I have attepmpted to build confidence in internal consistency of the proposed design method.

At the same time, I introduced Emergent Designer, an integrated reseach and design support tool that implements EED. In the first section of this chapter, I described the overall architecture of Emergent Designer and related individual components of the system to the phases of the proposed design method. The components of Emergent Designer were divided into three major groups: design components implementing the actual design method, analysis components offering various tools and methods for the analysis of the experimental results and design processes, and visualization components implementing various visualization methods and report generation tools. Also, a detailed description of the information flow within EED/Emergent Designer has been provided.

The second section of this chapter individually discussed each of the 10 components of Emergent Designer. In each case, the tasks performed by the component were listed and described. Also, the diagrams of the flow of information within the individual components were provided with detailed descriptions of the input/output relationships among components.

Finally, in the third section of this chapter, I discussed the actual implementation of Emergent Designer. It is a Java-based system with a fully-functional GUI that implements the proposed design method. It also integrates several open source and commercially available packages, e.g. *Mathematica* and *OpenOffice.org*, and communicates with them using available Java APIs. Implementation specific issues, i.e. algorithms, methods, functionality, etc., were discussed separately for each group of components of Emergent Designer.

The actual design experiments conducted using Emergent Designer are described in the following chapters.

6. DESIGN CONCEPT GENERATION USING CELLULAR AUTOMATA

"Order is not sufficient. What is required is something much more complex. It is order entering upon novelty; so that the massiveness of order does not degenerate into mere repetition; and so that the novelty is always reflected upon a background system"

(Alfred North Whitehead)

In this chapter, I begin the experimental part of this dissertation. I report results of various design experiments focused on generating novel design concepts of steel structural systems in tall buildings. In order to achieve this goal I utilize several types of concept generator mechanisms based on generative representations proposed earlier in chapter 4. The experiments described here have been conducted using Emergent Designer, an integrated research and design support tool introduced earlier in chapter 5.

The experimental results reported in this chapter constitute the first stage of the Empirical Performance Validation process, as discussed in section 3.6.3, in which the usefulness of the generative representations component of Emergent Engineering Design has been tested empirically for producing novel design concepts of wind bracing systems and the entire steel structural systems in tall buildings.

Figure 48 shows organization of this chapter. First, in introductory section 6.1, I revisit the research question 1 and the research hypothesis 1 (see section 3.3) and refine them in the context of the design problems considered in this dissertation. I also describe the types of experiments reported in this chapter. Next, sections 6.2 - 6.4 describe the results of the experiments in which design concepts of wind bracing systems in tall buildings were generated using types of generative representations based on cellular automata. Section 6.2 investigates the simplest generative representations based on elementary CAs and tests the impact of several representational parameters on the quality of generated design concepts.

Furthermore, in section 6.3, more complex types of generative represtations are studied involving two types of one-dimensional CAs: standard 1D CAs (subsection 6.3.1) and totalistic 1D CAs (subsection 6.3.2). Section 6.4 considers even more complex generative representations based on two-dimensional CAs. A particular emphasis in this case was put on explicit modeling of planar interactions among structural members by using various shapes and radii of the 2D local neighborhoods. As in the previous section, two types of 2D CAs were investigated: standard 2D CA (subsection 6.4.1) and totalistic 2D CA (subsection 6.4.2).

Finally, in section 6.5, I scale up the difficulty of the considered design problems and experimentally study design concept generators of the entire steel structural systems in tall buildings. I discuss the results of the experiments with generative representations consisting of multiple one-dimensional CAs (standard and totalistic) in which individual 1D CAs were employed to generate various subsystems of steel structures.

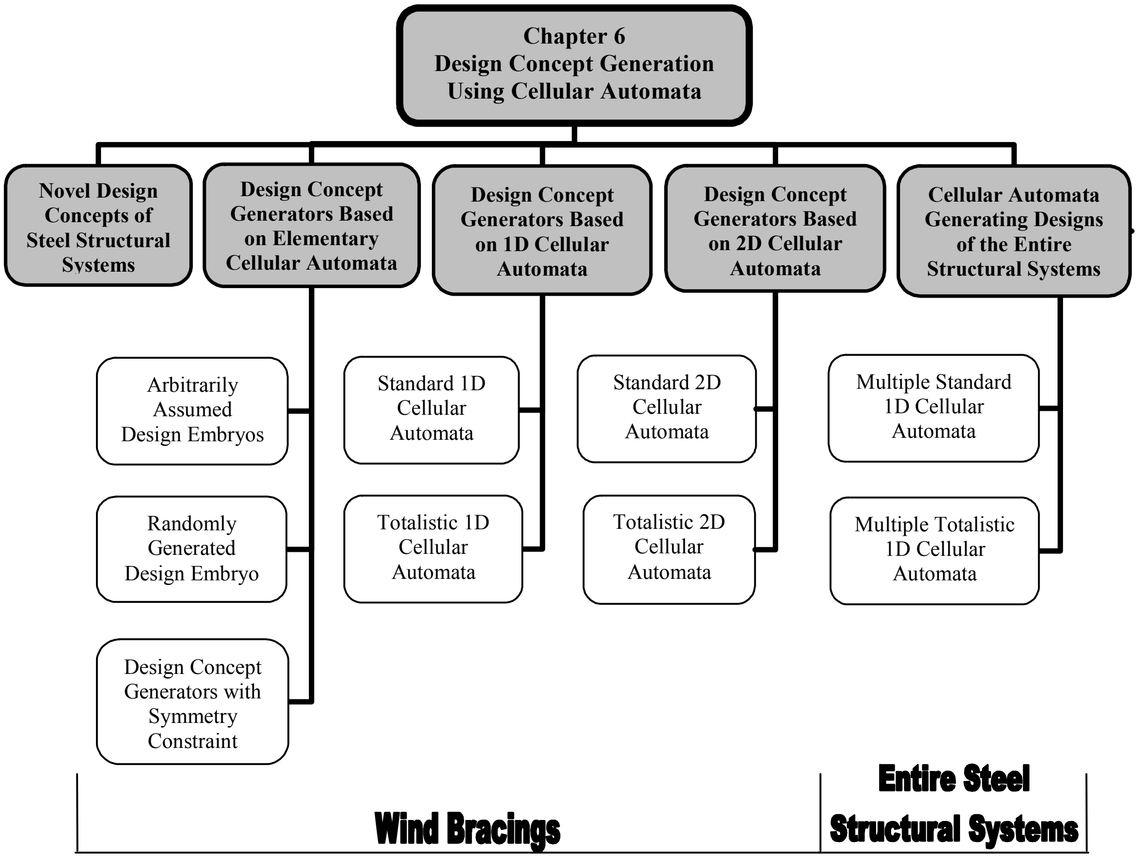


Figure 48. Organization of chapter 6

6.1. Novel Design Concepts of Steel Structural Systems

As stated earlier, in this chapter I describe results of the first stage of the Empirical Performance Validation process in which I empirically test the usefulness of the generative representations component of EED for producing novel design concepts. First, however, I need to define what I mean by a novel design concept in the context of conceptual design of steel structural systems in tall buildings. Earlier in section 2.1.2, I discussed the issue of creativity in design and provided several definitions of what makes a design concept creative. In this chapter, and in the remainder of this dissertation, I will employ the definition given by Gero (1996) who concludes that creativity in design "is not simply concerned with the introduction of something new into a design, although that appears to be a necessary condition for any process that claims to be labeled creative. Rather, the introduction of 'something new' should lead to a result that is unexpected (as well as being valuable)."

Thus, according to this definition, there are three important aspects of a novel design concept:

1. Something new
2. Something unexpected
3. Something valuable

In the context of conceptual design of steel structural systems in tall buildings we can translate these 3 conditions into the following criteria that a novel design concept must satisfy:

- It should be an unknown design concept.

- The introduced newness cannot be a mere random variation of a known design concept, or a design concept generated completely at random. On the contrary, a novel design concept must exhibit an unexpected structural shaping pattern.
- The value/quality of a design concept can be measured by its performance and feasibility. In the case of a steel structural system this performance can be measured by the total weight of a steel structure (a good estimate of its cost) and/or its maximum horizontal displacement (a good estimate of its stiffness).

Based on the discussion above, I can refine the research question 1 and the research hypothesis 1 in the specific context of a conceptual design of steel structural systems in tall buildings.

Research Question 1 (Refined):

Based on the existing knowledge on how to represent engineering systems; what mechanisms and models can be used to produce novel design concepts of steel structural systems in tall buildings?

Research Hypothesis 1 (Refined):

Evolutionary design and complex systems provide a framework for defining generative representations, i.e. representations of engineering systems based on simple programs, which can successfully produce novel design concepts exhibiting interesting structural shaping patterns and good performance in terms of the total weight of the structural systems and/or their maximum horizontal displacements.

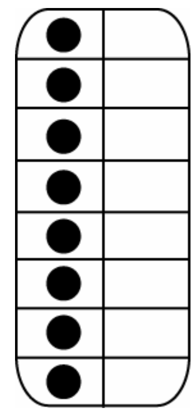
The refined research hypothesis 1 is more precise and can be tested empirically. The design experiments with generative representations of steel structural systems in tall buildings reported in this chapter were conducted to test this hypothesis. Also, the influence of some representation specific parameters on the quality of obtained design concepts was investigated experimentally.

In general, the experiments reported in this chapter can be classified using the parameters and their values shown in Table 4.

Table 4. Parameters and their values describing the types of experiments reported in this chapter

Design problem	Wind bracings	Entire steel structural system
Design Embryo	Arbitrary	Random
Embryo Location	Bottom	Top
Design Rule	1D CA	2D CA
Rule Type	Standard	Totalistic
Symmetry constraint	No	Yes
Boundary conditions	Periodic	Non-periodic
Analysis Type	First order	P-Delta

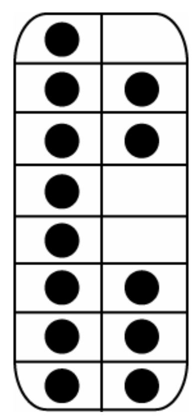
Generative representations for both design problems, i.e. design of a wind bracing system and design of an entire steel structural system in a tall building, were studied experimentally. The design concept generation mechanisms based on one-dimensional and two-dimensional CAs (described earlier in chapter 4) were used in the conducted experiments. Additionally, the influence of several representation specific parameters on the obtained results was investigated, including the location of the design embryo (top vs. bottom), the type of boundary conditions (periodic vs. nonperiodic), the type of the rule (standard vs. totalistic), and the way the design embryo is initialized (arbitrarily vs. randomly generated). Finally, I studied the possibility of adding specific domain knowledge, in this case the symmetry requirement which should improve designs' performance, and the impact of accuracy of the conducted structural analysis (first-order vs. P-Delta) on the quality of generated design concepts. The results of these parameter sensitivity studies were later considered in planning morphogenic evolutionary design experiments reported in chapter 8.



A small icon, similar to the one shown on the right, is placed at the beginning of each section of this chapter to indicate the values of the experimental parameters (defined in Table 4) which were used in the experiments reported in that section. For example, the icon shown on the right indicates that in the reported experiments the following values were used:

- design of a wind bracing system was considered,
- arbitrarily assumed design embryos were employed,
- design embryos were located at the bottom of the structural system,
- design rules based on standard 1D CA rules were used,
- no symmetry constraint and periodic boundary conditions were imposed,
- structural analysis was conducted using the first-order analysis only.

6.2. Design Concept Generators Based on Elementary Cellular Automata



In this section, I begin with the simplest possible generative representation of a wind bracing system proposed earlier in section 4.4.1. I investigate one-dimensional cellular automata with only 2 possible cell values and the local neighborhood of size 3. These CAs are commonly called elementary CAs (see section 2.2). Elementary CAs were used to generate design concepts of wind bracing systems in tall buildings. The design concept generation mechanism used here is based on the generative representation consisting of a single design embryo and a single design rule. In the conducted experiments, the concept generation mechanisms based on elementary CAs defined the topologies/configurations of a wind bracing system in a tall building. On the other hand, the topologies/configurations of the beam system and the column system in a tall building were arbitrarily assumed and kept the same in all experiments.

When the topology/configuration of a wind bracing system was defined, the sizing optimization algorithm implemented in SODA and described in (Grierson 1989) was used to determine the optimal cross-sections of structural members in the entire steel structural system. In other words, the sizing optimization was conducted not only for the wind bracing elements but also for beams and columns. The optimal cross-sections of structural members were selected from the catalog of standard shapes specified in (American Institute of Steel Construction 1989). An arbitrary assumption was made, motivated by manufacturability issues, that the cross-sections of each type of structural members were allowed to change every three stories in a steel structural system. In other words, cross-sections of each type of structural members were assumed the same if the members were located within the same 3-story segment of the steel structural system. For example, all K bracings located from story 1 to story 3 were assigned the same cross-sections, all K bracings located from story 4 to story 6 had the same cross-sections that could be, in general, different than the cross-section of K bracing located below, etc.

In the structural analysis conducted by SODA, dead, live, and wind loads were considered. The magnitudes of loads used in the design experiments reported in this dissertation are provided in Table 5. Five load combinations were considered, following the design specifications for steel, concrete, and composite structures in tall buildings given in (Taranath 1998). They included the following combinations of loads:

- Dead + Live
- 0.75(Dead + Live + Wind)
- 0.75(Dead + Live – Wind)
- 0.75(Dead + Wind)
- 0.75(Dead – Wind)

The negative sign placed in front of the wind loads indicates that the wind forces considered in a given load combination act in the opposite direction, i.e. wind pressure is replaced by wind suction and vice versa, when compared to the case when the plus sign is used.

Table 5. Magnitudes of dead, live, and wind loads used in design experiments

Load Parameter	*Value(s)*
Dead load magnitude	50 psf (2.39 kN/m^2)
Live load magnitude:	
- building	100 psf (4.78 kN/m^2)
- roof	30 psf (1.43 kN/m^2)
Wind load:	
- Wind speed	100 mph (160.9 km/h)
- Wind importance factor	1.0
- Wind exposure category	C

In SODA, the structural analysis can be conducted using either first order or P-Δ analysis. In the experiments reported in this section, both types of structural analysis were performed and the differences in obtained values subsequently compared.

When the topology of a wind bracing system has been defined and optimal cross-sections of all structural members have been computed, the total weight of the steel structure and the maximum horizontal displacement (sway) of the steel structure (measured at the top rightmost node of the structural system as shown in Figure 49) were calculated. This sizing optimization process was conducted without imposing any maximum displacement constraints (serviceability conditions). The obtained values are reported in this section together with the topologies of the steel structural systems.

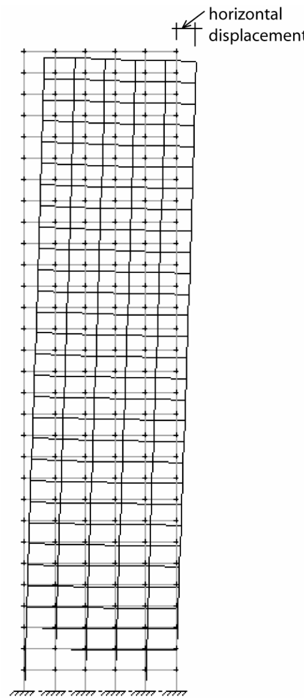


Figure 49. Sway: measuring maximum horizontal displacement of a structural system

The values of the parameters defining the domain, as discussed earlier in section 5.2.1, are shown in Table 6. The majority of the parameters included in this table will have the same values in all design experiments reported in this dissertation.

Table 6. Domain parameters and their values used in the reported experiments

Domain Parameter	*Value(s)*
Design code	AISC-LRFD-93
Problem dimensionality	2D
Design type	Frame
Behavior (analysis) type	First-order, or P-Δ
Sidesway	Permitted
Cross-sections database	AISC
Unit system	U.S. customary
Length unit	ft
Force unit	lbs

Table 7 shows the parameters of the design problem considered in this section. As discussed earlier, elementary CAs, i.e. CAs with two cell state values and the neighborhood of size three, were used to generate design concepts of wind bracing systems in tall buildings. Thus, two types of wind bracing elements could be used in each design experiment utilizing elementary CA. In the reported experiments, two groups of wind bracing elements were employed. The group No.1 included no bracing (empty cell) and simple X bracing while the group No.2 contained no bracing and K bracing.

Table 7. Problem parameters and their values used in experiments with elementary CAs

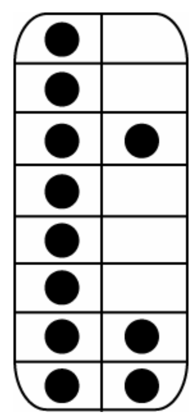
Problem Parameter	*Value(s)*
Problem type	Design of a wind bracing system in a tall building
Number of stories	30
Number of bays	5
Bay width	20 feet (6.01 m)
Story height	14 feet (4.27 m)
Distance between transverse systems	20 feet (6.01 m)
Types of bracing elements	None and simple X bracings, None and K bracings
Types of beam elements	Fixed-Fixed beams (only)
Types of column elements	Fixed-Fixed columns (only)
Types of supports	Fixed supports (only)

The design experiments with elementary CAs were divided into three parts:

1. Experiments with arbitrarily assumed design embryos.
2. Experiments with randomly generated design embryos.
3. Experiments with the symmetry constraint. In this case, the symmetry constraint was imposed on the design rules which were subsequently applied to a set of symmetric design embryos.

The obtained results are discussed in the following subsections.

6.2.1. Arbitrarily Assumed Design Embryos



In the first group of experiments utilizing elementary CAs as design concept generators, arbitrarily assumed design embryos were used. The design embryos consisted of five cells due to the fact that 5-bay buildings were considered here (see problem parameters in Table 7). Depending on the group of types of wind bracing elements used in the experiments, either group No.1 (no bracings and simple X bracings), or group No.2 (no bracings and K bracings), the central cell in the design embryo had a value representing either simple X bracing (group No.1) or K bracing (group No.2). The remaining four cells had values representing no bracings (for both groups), as it is shown in Figure 50.

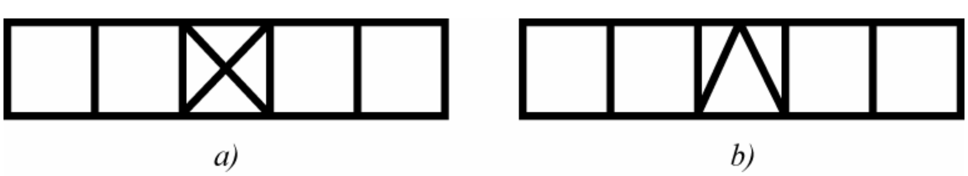


Figure 50. Design embryos iterated by elementary CAs when bracings from a) the group No. 1, b) the group No. 2 are used

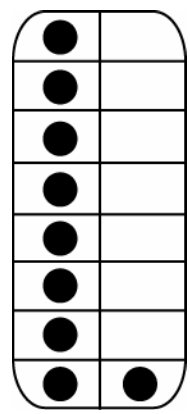
There are 256 design rules based on elementary CAs (see explanations in section 2.2). All of them were applied to the design embryos and iterated the number of times which is one less than the number of stories in a tall building, i.e. 29 times for 30-story buildings considered here. In this way the entire configuration of a wind bracing system was developed from the design embryo shown in Figure 50a), or Figure 50b), using the corresponding design rule. When the topology of the wind bracing system was defined, the sizing optimization was performed as discussed in the previous section. Finally, the total weight of the steel structure and its maximum horizontal displacement were calculated using both first-order analysis and P-Δ effects.

The impact of several representation specific parameters on the quality of obtained design concepts was tested experimentally in this section, including the following parameters:

- Location of the design embryo (bottom vs. top of a structural system)
- Type of boundary conditions (periodic vs. nonperiodic)

Initial experiments considered elementary CA rules with periodic boundary conditions and the design embryo located at the bottom of a structural system. Next, the same elementary CA rules were employed but this time, the design embryo was located at the top. Finally, the third set of experiments investigated elementary CA rules with nonperiodic boundary conditions. The results of these experiments are reported in the following subsections.

Design Embryo at the Bottom and Periodic Boundary Conditions



The experiments reported in this subsection involved the design embryo (see Figure 50) located at the bottom of a steel structural system and periodic boundary conditions (see section 4.4.1). The results of all these experiments, i.e. the complete set of all 256 design concepts of wind bracings systems, are presented in Appendix B.

Table 8 shows only the 12 best designs with respect to the total weight of the steel structural systems obtained in these experiments. Each cell in Table 8 contains the number of a design rule at the top (see the explanation of the numbering scheme of cellular automata rules in section 2.2.2), the actual design developed from the design embryo by this rule (center), and four values arranged in 2 x 2 array (the bottom part) as shown on the right. This array contains four values representing the total weight of the steel structural system (first row) and its maximum horizontal displacement (second row). The first column contains measurements obtained using the first-order structural analysis while the second column contains the values produced by a more accurate and at the same time more computationally expensive P-Δ analysis. The values of the total weight of the steel structural system presented in the first row are measured in lbs. whereas the values of the maximum horizontal displacement, shown in the second row, are measured in inches.

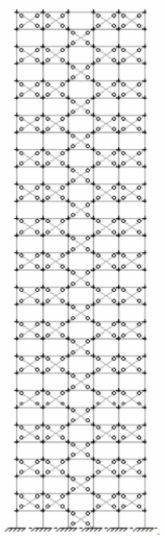
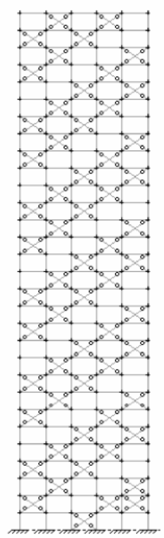
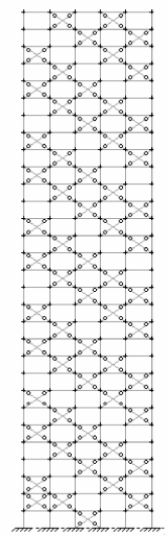
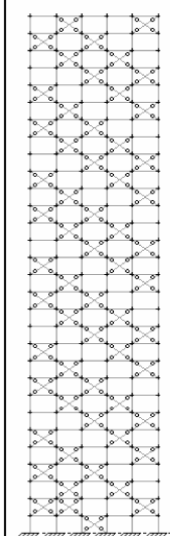
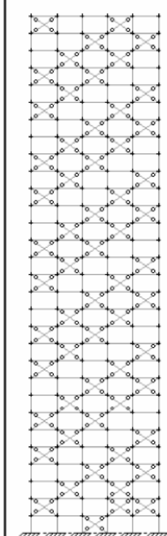
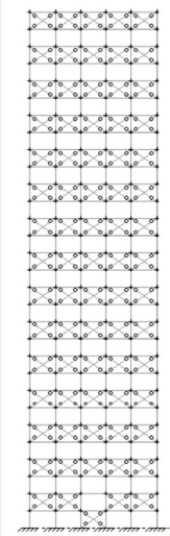
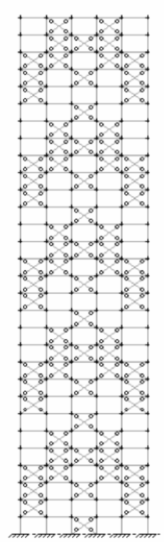
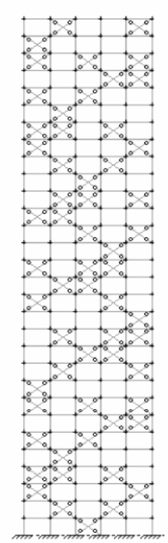
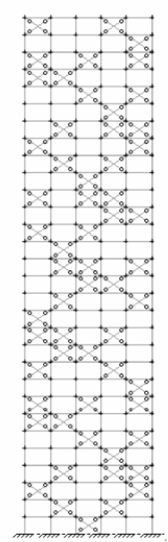
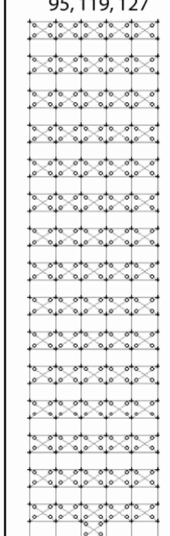
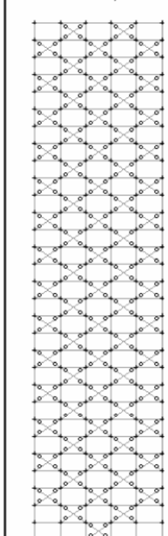
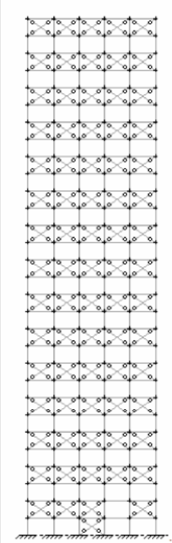
Total weight of the steel structure (1st-order)	Total weight of the steel structure (P-Delta)
Maximum horizontal displacement (1st-order)	Maximum horizontal displacement (P-Delta)

All 256 design concepts of wind bracing systems presented in Appendix B were generated by the simplest possible design rules represented by elementary CA rules.

Structural Shaping Patterns

Appendix B shows a great diversity of structural shaping patterns generated by the design rules based on elementary CA rules. All these diverse patterns were created using the simplest possible design embryo (single simple X bracing located in the central bay) and simplest possible design rules (elementary CA rules). Thus, there is a great potential for developing novel designs of structural systems even using this simplest type of generative representations. In fact, the set of 256 design concepts in Appendix B contains 144 unique structural shaping patterns of wind bracing systems. Some shaping patterns found in Appendix B repeat due to the simplicity of the arbitrarily chosen design embryo and imposed periodic boundary conditions.

Table 8. Best designs in terms of the total weight of the steel structural system (calculated using the P-Δ analysis) produced by elementary CA rules with periodic boundary conditions and the design embryo located at the bottom

Rule 51  560,646 562,570 4.9940 5.1378	Rule 177  569,840 565,145 4.6401 4.7913	Rule 163  571,001 566,306 4.6689 4.8206	Rule 99  566,738 566,738 4.6901 4.7891	Rule 57  566,738 566,871 4.6311 4.7283	Rule 19  565,214 567,494 6.6620 6.8912
Rule 105  567,245 568,330 4.7104 4.7826	Rule 82, 210  567,006 568,671 5.1936 5.3053	Rule 26, 154  568,563 569,488 5.2300 5.3657	Rule 23,31, 55, 63, 87, 95, 119, 127  565,385 569,840 6.6824 6.8606	Rule 50, 178  568,951 570,251 4.3405 4.4116	Rule 7  567,943 570,439 6.6236 6.8527

Several interesting structural shaping patterns found in Appendix B are presented in Table 9. A few design concepts generated by the design rules based on elementary CAs and shown in Table 9 are similar, and sometimes even identical, to the shaping patterns known from the structural engineering literature and presented in Table 10. For example, the structural pattern developed by rules 4, 12, etc. shown in Table 9 is identical to the concept of a vertical truss (see Design 6 in Table 10). Similarly, the pattern generated by rules 151, 159, etc. presented in Table 9 is similar to the concept of a fully braced frame (see Design 1 in Table 10). As it is shown in Table 9, elementary CAs can produce not only shaping patterns known from the structural engineering literature (first four designs in row 1 in Table 9) but also many novel configurations of bracing elements that exhibit good performance.

Table 9. Interesting structural shaping patterns produced by elementary CA rules with periodic boundary conditions and the design embryo located at the bottom

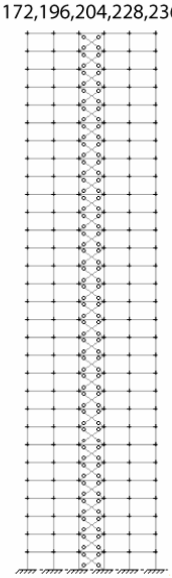
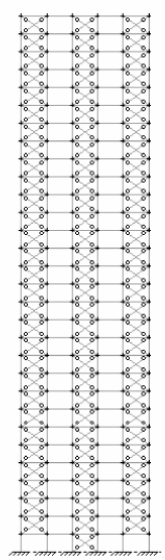
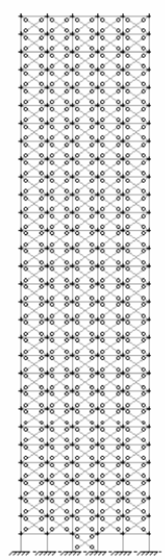
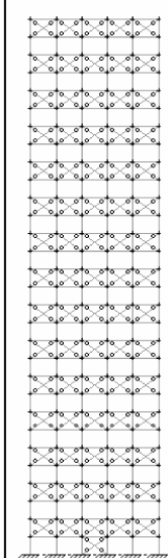
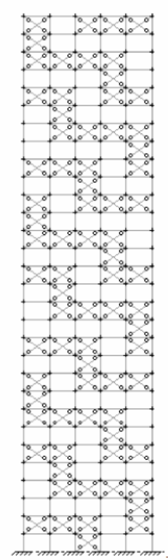
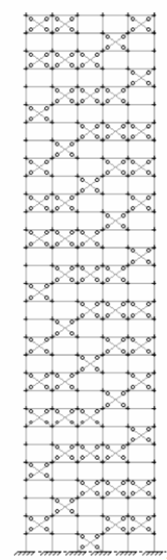
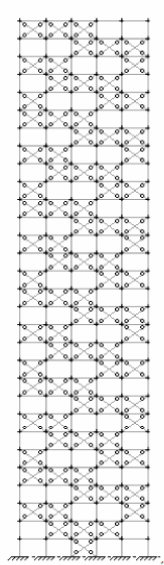
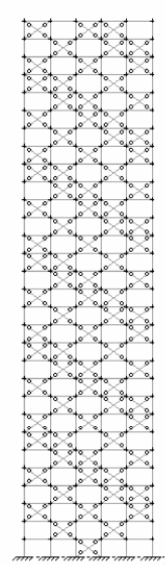
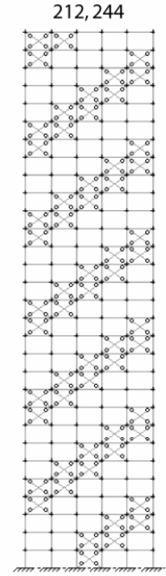
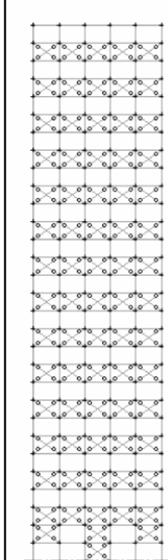
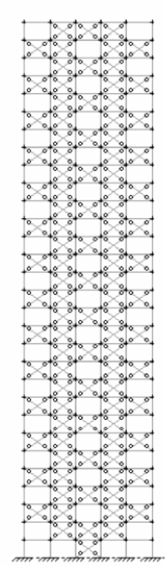
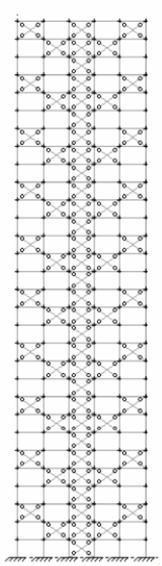
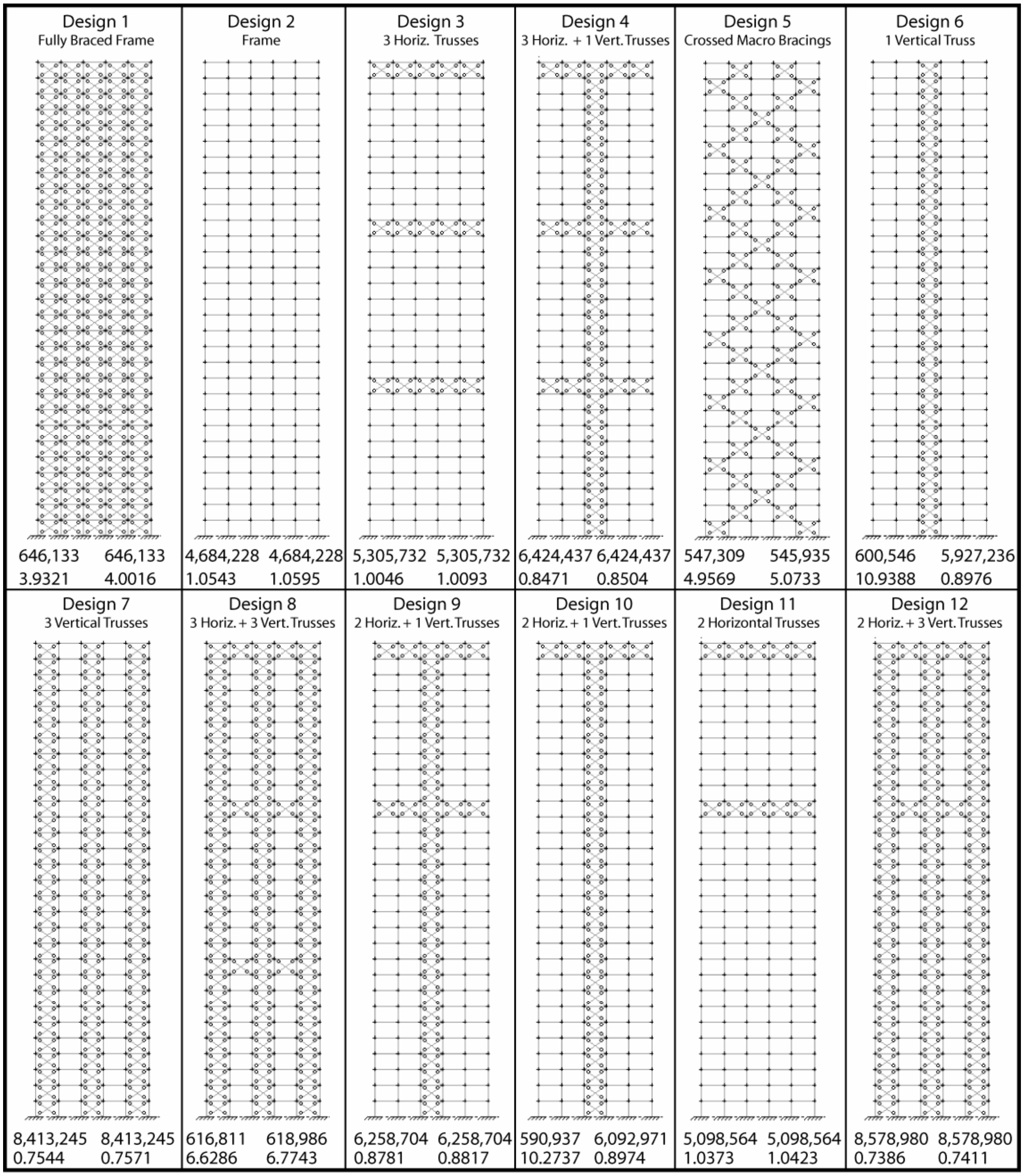
Rule 4,12,36,44,68,76, 100,108,132,140,164, 172,196,204,228,236	Rule 77, 205	Rule 151,159,183, 191,215,223,247,255	Rule 23, 31, 55, 63, 87, 95, 119, 127	Rule 15	Rule 17, 49
					
600,546 5,927,236	8,330,379 8,330,379	648,338 648,338	565,385 569,840	581,884 588,157	573,363 573,363
10.9388 0.8976	0.7645 0.7672	3.9057 3.9739	6.6824 6.8606	5.2233 5.2680	5.2786 5.4134
Rule 30, 62	Rule 58	Rule 84, 116, 212, 244	Rule 109	Rule 126	Rule 150
					
578,539 584,644	585,202 8,288,946	584,033 584,432	579,603 580,977	602,305 602,305	578,577 579,168
4.6663 4.7025	4.2916 0.6659	5.3193 5.4434	6.1875 6.4100	4.3327 4.4193	5.0589 5.1561

Table 10. Examples of design concepts of wind bracing systems from the structural engineering literature



Classification of Structural Shaping Patterns

One can attempt to categorize structural shaping patterns in many ways, e.g. in terms of the quality of produced design concepts, their physical appearance, etc. A classification presented below exploits the dynamical properties of the design concept generation mechanism based on cellular automata. It divides the structural shaping patterns into four distinct classes, based on four classes of dynamical behavior of cellular automata. This classification was initially proposed by Wolfram (1983) (see section 2.2.1 for more details) and has been adapted to structural design problems in this dissertation.

This classification of structural shaping patterns is presented in Table 11. It is based on four distinct classes of dynamical behavior of cellular automata that generate shaping patterns: fixed-point behavior, periodic behavior, apparently ‘chaotic’ behavior, and localized propagating structures. The first class (fixed-point behavior or class 1) includes uniform patterns produced by the design rules 0, 4, 8, etc. (see Table 11). These patterns correspond to the fixed-point behavior in dynamical systems, i.e. configurations of wind bracings at subsequent stories in a tall building (eventually) converge to an identical configuration of wind bracings. The second class (periodic behavior or class 2) consists of periodically repeating patterns generated by the design rules 1, 23, 33, etc. Repetition periods of the shaping patterns vary depending on the design rule used. This group of patterns corresponds to periodic/cyclic behavior in dynamical systems, i.e. configurations of wind bracings at subsequent stories in a tall building (eventually) repeat with a constant repetition period.

The third class (apparently ‘chaotic’ behavior or class 3) includes shaping patterns of apparent irregularity/randomness produced by the design rules 26, 82, 154, etc. This group can be related to the chaotic behavior produced by some classes of dynamical systems. Of course, in the case of structural shaping patterns generated by the simplest elementary CA, the actual chaotic behavior cannot be obtained due to discreteness and finiteness of this design space. On the other hand, the shaping patterns generated by rules 26, 82, etc. have no apparent regularity or periodicity, as it was the case with the patterns discussed earlier. Hence, they have been placed in class 3 because the shaping patterns are characterized by apparent irregularity/randomness.

Interestingly, these apparently ‘chaotic’ structural shaping patterns exhibit very good performance. For example, the shaping pattern generated by rule 82 (and at the same time by rule 210) is the 8th best design concept (see Table 8) with respect to the total weight of the steel structural system.

Finally, the last group consists of shaping patterns characterized by localized and propagating structures, e.g. as the ones generated by the design rules 2, 14, 38, 46, 57, etc. This group corresponds to so-called localized propagating structures behavior in Wolfram’s hierarchy. From the structural design point of view, the localized propagating structures, shown in Table 11 (and also many others in Appendix B), form emergent concepts of so-called macro bracings, or super diagonals. The macro bracings shown in Table 12 have various widths measured by the number of adjacent bracing elements forming the macro bracing. For example, the macro bracing generated by the rule 2, 10, etc. has the width equal to 1 bracing, the one produced by rule 14, 46, etc. has the width equal to 2 bracings. Rule 143 generates the macro bracing of width equal to 3 bracings and rule 175 produces one with the width equal to 4 bracings.

Another interesting design concept of a macro bracing emerges from the design rules 177, 163, 99, 57 (see Table 8 and Table 12) in which the macro bracing pattern has the width of 2 but the adjacent bracing elements are located at a distance equal to one story. In fact, these design concepts exhibit very good performance in terms of both the total weight of the steel structural system and its maximum horizontal displacement (see Table 8). Finally, an interesting macro bracing pattern is generated by rule 227 (see Table 12) in which 2 macro bracings, one of width 1 and the other of width 2, are interwoven.

Table 11. Categorization of structural shaping patterns based on four classes of dynamical behavior of cellular automata

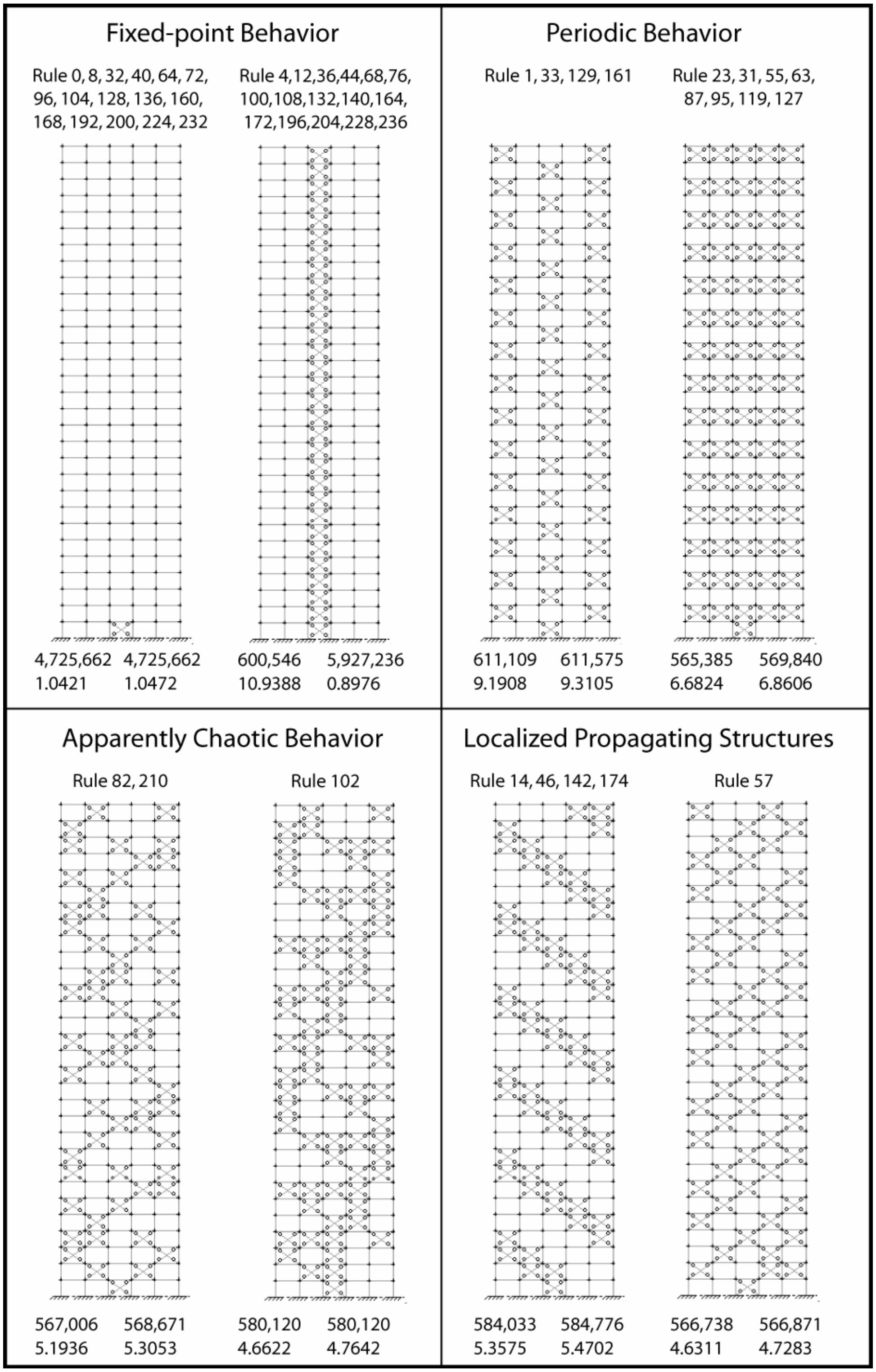
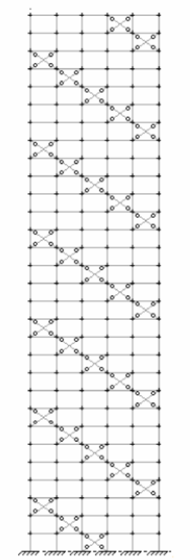
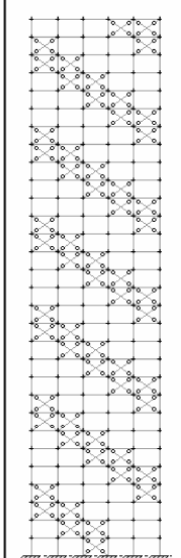
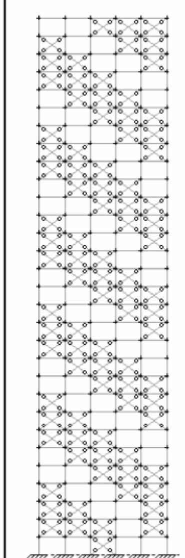
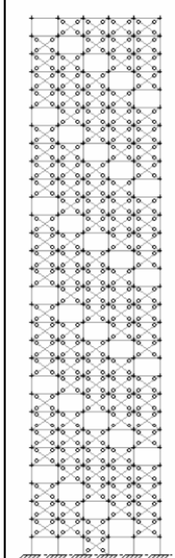
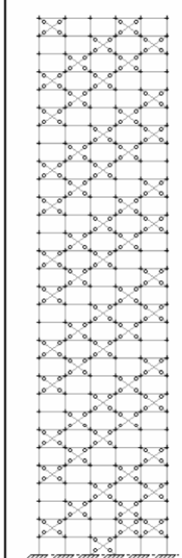
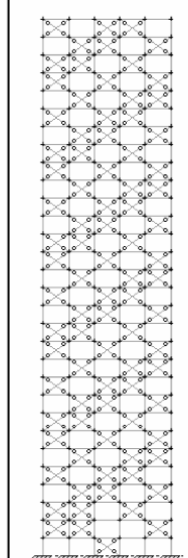


Table 12. Emergent shaping patterns of macro bracings (super diagonals) generated by elementary cellular automata

Rule 2, 10, 34, 42, 66, 74, 98, 106, 130, 138, 162, 170, 194, 202, 226, 234	Rule 14, 46, 142, 174	Rule 143	Rule 175	Rule 57	Rule 227
					
5,927,236 5,927,236 0.8087 0.8116	584,033 584,776 5.3575 5.4702	604,130 602,082 4.5504 4.6498	616,673 616,673 4.1147 4.1904	566,738 566,871 4.6311 4.7283	584,469 584,469 4.2896 4.3730

As is it shown in Table 8, the design concepts with emergent macro bracings form 2nd, 3rd, 4th, and 5th best designs generated by elementary CA rules. The overall best design (with respect to the total weight of the structural system) produced by elementary CA rules is generated by rule 51. It develops another interesting structural shaping pattern consisting of two subpatterns: the emergent pattern of crossed diagonal bracings located in the center of the structural system and the emergent pattern of horizontal trusses located every two stories in a structural system.

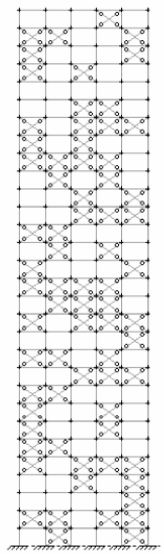
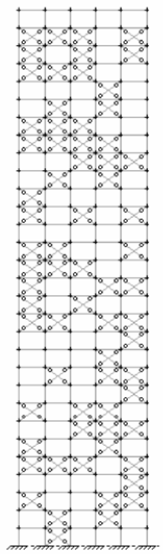
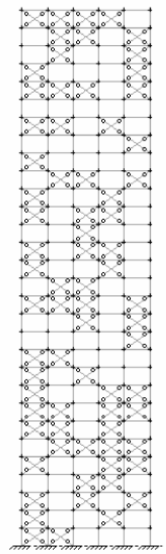
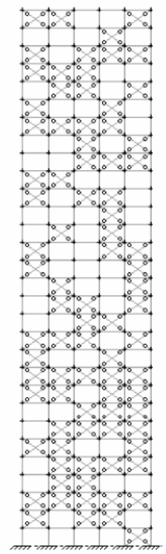
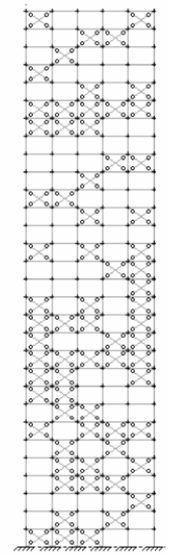
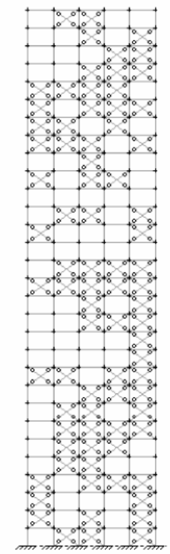
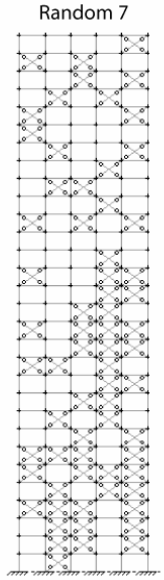
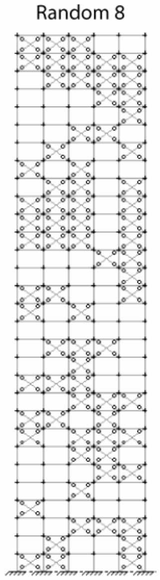
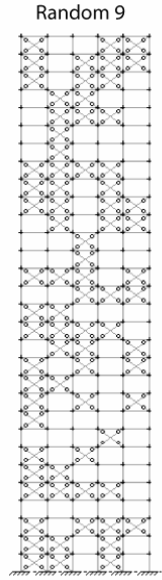
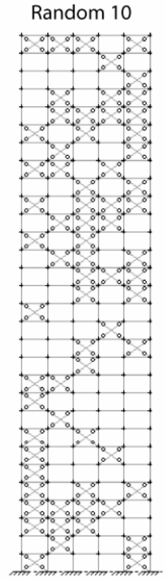
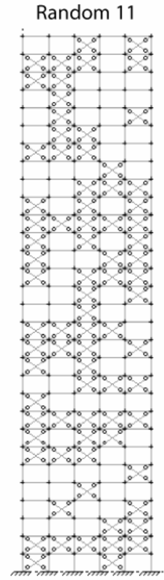
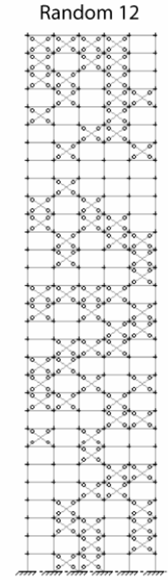
Rule 105 generates an intriguing structural shaping pattern with good performance (see 7th best design in Table 8). Previously mentioned rules 82, 210, 26, and 154 produce mirrored design concepts of apparently 'chaotic' patterns which exhibit good performance (8th and 9th best). Finally, rules 50 and 178 develop an interesting macro pattern (a type of a checkerboard pattern) which proves to perform well in terms of the total weight and the maximum horizontal displacement of the steel structural system.

Elementary Cellular Automata vs. Randomly Generated Designs

The design concepts shown in Appendix B were also compared qualitatively and quantitatively with randomly generated configurations of wind bracings systems. In order to do that a comparable sample of configurations of wind bracing systems was randomly generated and evaluated using the first-order structural analysis. Examples of 12 designs (out of 250) from this randomly generated sample are shown in Table 13.

It is clear that there are large qualitative differences between designs generated by elementary CA rules (see Table 8 and Appendix B) and randomly generated ones (see Table 13). In the quantitative comparison of both samples, i.e. the sample of designs generated by elementary CAs and the sample generated randomly, the total weight of the structural systems and their maximum horizontal displacement calculated using the first-order analysis were considered.

Table 13. Examples of randomly generated design concepts of wind bracing systems in tall buildings

Random 1	Random 2	Random 3	Random 4	Random 5	Random 6
					
590,510 592,685 5.1769 5.2749	593,091 595,741 5.2012 5.2567	587,810 588,309 5.2676 5.4091	599,735 600,068 4.6057 4.7010	600,255 602,210 5.4808 5.6543	597,681 601,766 5.3962 5.5488
Random 7	Random 8	Random 9	Random 10	Random 11	Random 12
					
587,023 591,679 5.5129 5.5552	7,791,743 7,791,743 0.7159 0.7182	598,228 598,528 5.2420 5.3734	592,245 7,501,709 5.1153 0.7450	7,957,478 618,693 0.6897 4.9349	571,922 575,051 5.4660 5.4733

In the analysis of the distribution of the total weight of generated structural systems, only feasible design concepts were considered. The design concept was included in the group of feasible designs if its total weight was less than 1,000,000 lbs. (this value was arbitrarily assumed). The number of designs generated by elementary CAs satisfying this criterion was equal to 172 while the number of randomly generated designs satisfying the same criterion was equal to 221. The dotplot displaying both sample distributions is presented in Figure 51. Basic descriptive statistics for both sample distributions are shown in Table 14.

Figure 51 and Table 14 show that the two distributions differ. Elementary CAs produce structural shaping patterns that exhibit better performance (in terms of the reduced total weight) compared to randomly generated design concepts. In fact, no randomly generated design concept has the total weight smaller than 570,000 lbs. while there are 22 such designs in the sample generated by elementary CAs.

Advantages of elementary CAs in producing design concepts of better performance are further supported by the analysis of the distributions of the maximum horizontal displacement for both sample distributions. Figure 52 and Table 15 provide qualitative and quantitative comparison of both sample distributions. Again, elementary CAs generate structural shaping patterns that exhibit better performance with respect to maximum horizontal displacement than the design concepts generated randomly.

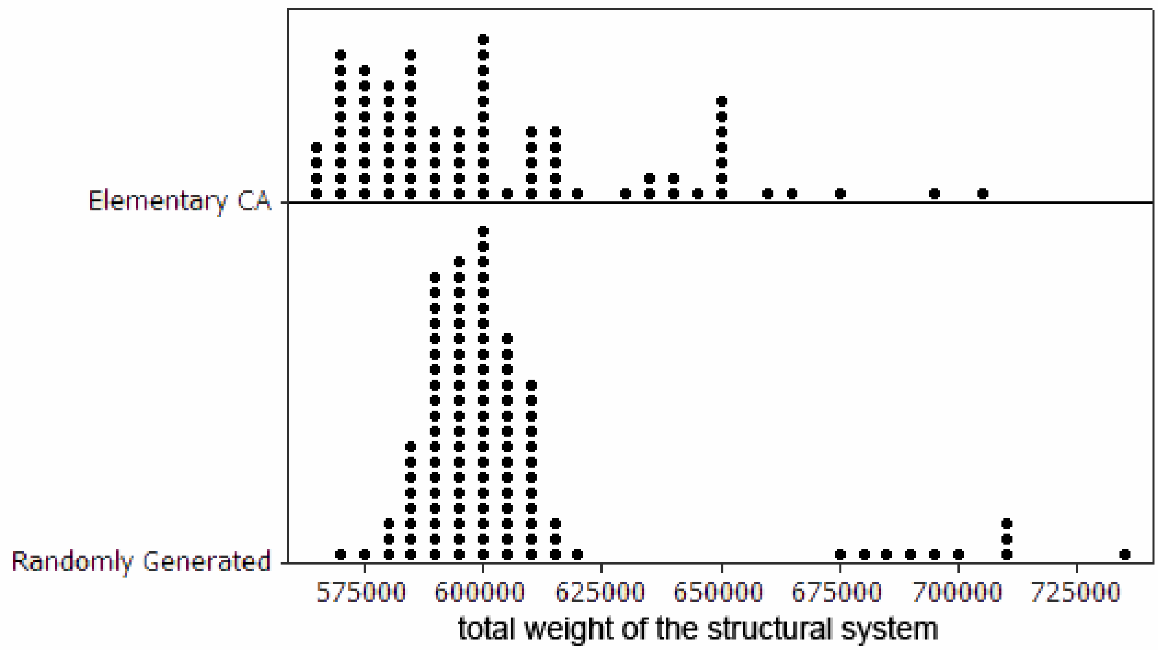


Figure 51. Dotplot comparing the distributions of the total weight of steel structural systems generated by elementary CAs and generated randomly

Table 14. Descriptive statistics summarizing the sample distributions of the total weight of structural systems generated by elementary CAs and generated randomly

Quantity	*Elementary CA*	*Randomly Generated*
Sample size	172	221
Minimum weight	560,646	571,922
Maximum weight	707,354	733,310
Mean weight	598,517	605,285
Median weight	590,875	598,269
Standard Deviation	29,247	28,091

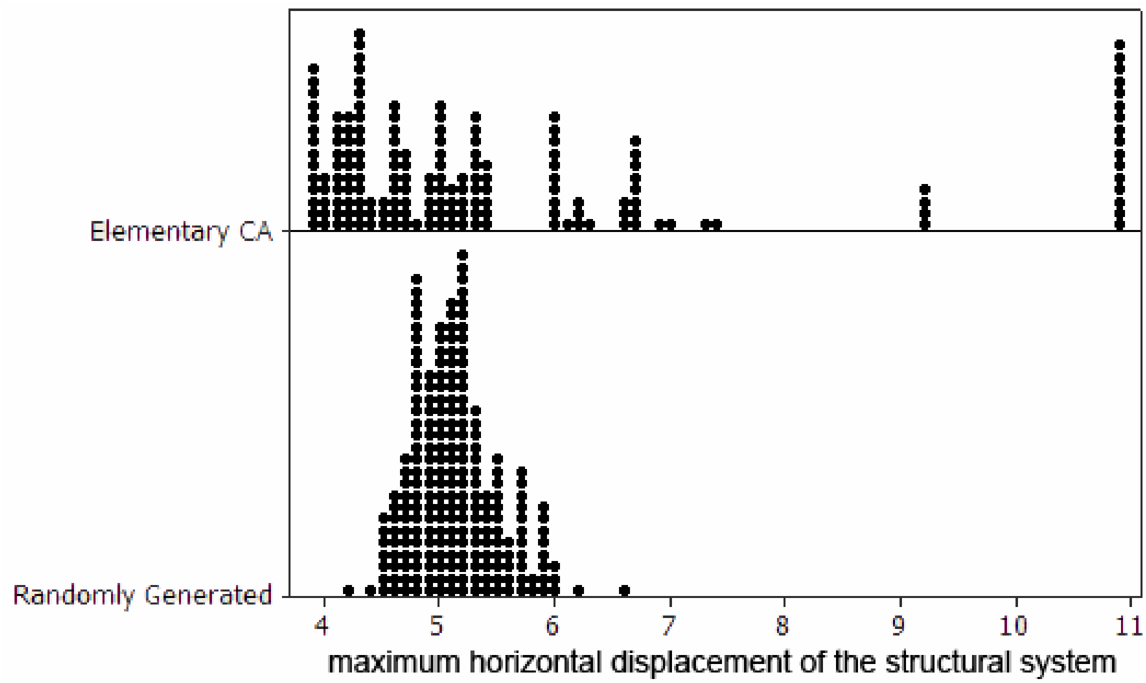


Figure 52. Dotplot comparing the distributions of the maximum horizontal displacement of steel structural systems generated by elementary CAs and generated randomly

Table 15. Descriptive statistics summarizing the sample distributions of the maximum horizontal displacement of structural systems generated by elementary CAs and generated randomly

Quantity	*Elementary CA*	*Randomly Generated*
Sample size	172	221
Minimum displacement	3.892	4.2407
Maximum displacement	10.939	6.5566
Mean displacement	5.593	5.1296
Median displacement	4.955	5.1075
Standard Deviation	0.153	0.0256

Finally, Figure 53 compares both distributions with respect to the two objectives, i.e. the total weight of structural systems and their maximum horizontal displacements. It is clear that the design concepts produced by elementary CAs are better than the design concepts produced randomly with respect to both objectives. In other words, using the multiobjective optimization terminology (see section 2.1.5), design concepts generated by elementary CAs dominate the ones produced randomly.

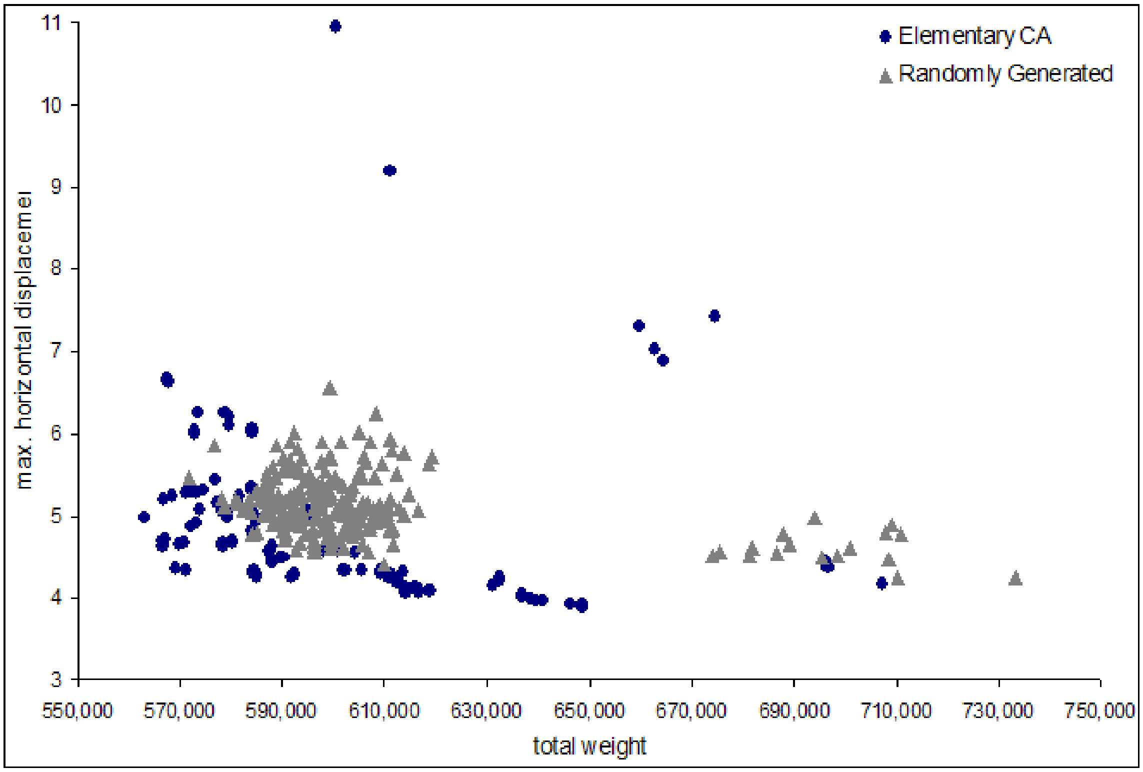


Figure 53. Comparison of both sample distributions with respect to two objectives: the total weight of steel structures (horizontal axis) and their maximum horizontal displacements (vertical axis)

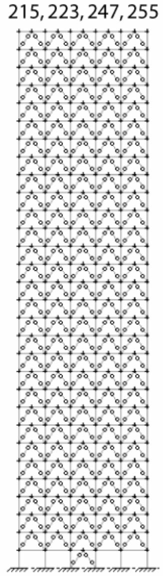
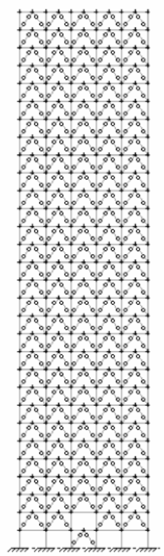
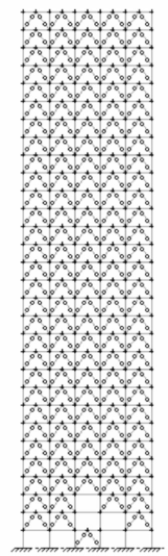
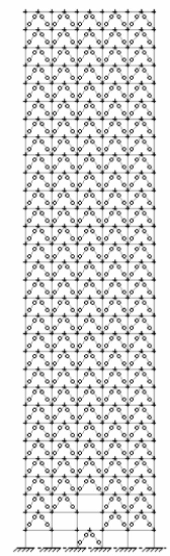
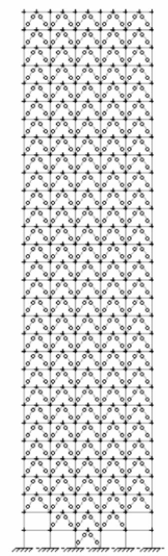
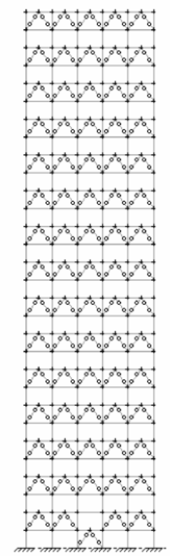
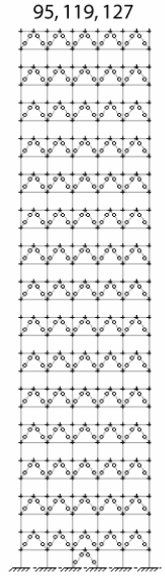
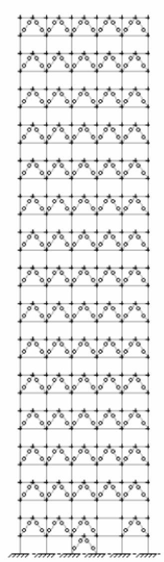
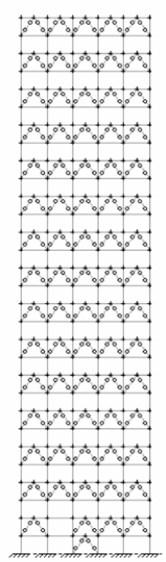
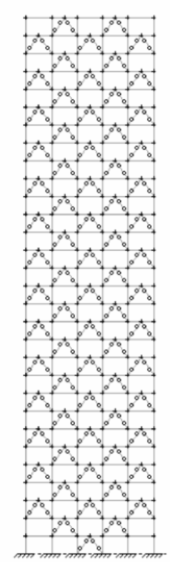
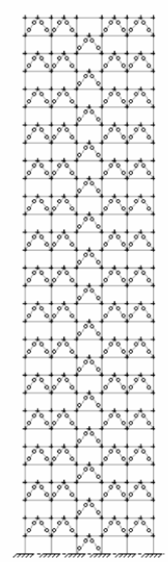
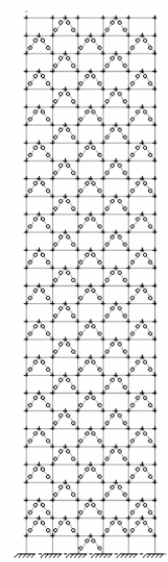
K Bracings

The design experiments reported above were repeated with the second group of bracing elements consisting of K bracings and no bracings (empty cells). In these experiments, the design embryo shown in Figure 50b) was used. As before, it was located at the bottom of a steel structural system. Also, periodic boundary conditions were imposed. As in the previous experiments, the entire set of 256 elementary CA rules was employed to develop design concepts of wind bracing systems. Table 16 shows the best designs (in terms of the total weight calculated using the P-Δ analysis) obtained in these experiments.

The twelve best design concepts shown in Table 16 exhibit four distinct structural shaping patterns, including:

- fully braced pattern (designs 1-5), similar to the one in Design 1 shown in Table 10
- pattern of horizontal trusses located every two stories (designs 6-9)
- checkerboard pattern (designs 10 and 12)
- combined pattern consisting of crossed diagonal bracings located in the center of the structural system and horizontal trusses located every two stories (design 11)

Table 16. Best designs of wind bracing systems consisting of K bracings produced by elementary CA rules with periodic boundary conditions and the design embryo located at the bottom

Rule 151,159,183,191, 215, 223, 247, 255	Rule 251	Rule 235	Rule 249	Rule 222, 254	Rule 19
					
450,234 453,021 4.9863 5.0781	457,501 459,519 4.9596 5.0551	459,205 461,223 4.9467 5.0418	459,205 461,223 4.9562 5.0515	459,513 461,960 4.9326 5.0204	497,106 498,121 7.5449 7.7694
Rule 23, 31, 55, 63, 87, 95, 119, 127	Rule 7	Rule 21	Rule 50, 178	Rule 51	Rule 179
					
494,768 498,248 7.5612 7.7233	497,047 498,939 7.5183 7.7692	497,047 498,939 7.5273 7.7784	502,854 502,854 5.5106 5.6518	501,044 503,376 6.2768 6.4285	505,655 505,796 5.4226 5.5571

Simple X bracings vs. K Bracings

Table 16 also shows that a significant reduction of the total weight of the produced structural systems can be obtained when K bracings are used instead of simple X bracings. At the same time, however, the stiffness of structural systems is reduced as they exhibit larger horizontal displacements. These observations have been further confirmed by simple statistics reported in Table 17. In this table, design concepts developed using the same elementary CA rules were compared, i.e. design concepts with identical structural shaping patterns which differed only in the group of bracing elements used (group No.1 or group No.2). In other words, pairs of design concepts (one with simple X bracings and one with K bracings) developed using identical elementary CA rules were compared in terms of the total weight of the generated structural system and its maximum horizontal displacement. Then, based on the results of these comparisons, some simple statistics were calculated.

Table 17. Comparison of design concepts generated by elementary CA rules utilizing simple X bracings and K bracings

Quantity	*1st order*	*P-Δ*
Number of design concepts with reduced weight	192	203
Number of design concepts with increased horizontal displacement	173	187
Median weight reduction	72,896 lbs.	77,055 lbs.
Median percentage of weight reduction	13.0%	14.3%
Median displacement increase	0.8484 in.	0.9393 in.
Median percentage of displacement increase	13.6%	17.1%

Table 17 shows that 192 out of 256 design concepts (75 percent), or 203 out of 256 design concepts (79 percent) in the case of the P-Δ analysis, developed using elementary CA rules with K bracings have reduced total weight compared to the same design concepts constructed with simple X bracings. At the same time, 173 design concepts (67 percent) consisting of K bracings, or 187 (73 percent) in the case of the P-Δ analysis, have increased horizontal displacements compared to design concepts constructed with simple X bracings. When K bracings are used in place of simple X bracings, the median reduction of the total weight of the structural systems from the sample of 256 designs developed using elementary CA rules is larger than 70,000 lbs. This corresponds to 13 percent (median) reduction of steel consumption. On the other hand, the use of K bracings instead of simple X bracings causes a median increase of horizontal displacement of about 0.85 inch, or 13.6 percent.

Elementary Cellular Automata vs. Randomly Generated Designs

Further, 256 design concepts developed by elementary CA rules were compared to a comparable sample of 250 randomly generated designs with respect to two objectives, i.e. the total weight of the steel structural system and its maximum horizontal displacement, as it is shown in Figure 54. It shows only design concepts whose total weight was less than 800,000 lbs., i.e. 197 design concepts developed by elementary CA rules, and 232 designs generated randomly.

Also in this case, design concepts developed by elementary CA rules dominate the ones generated randomly with respect to both objectives. One can easily recognize in Figure 54 three distinct regions in this performance space. On the left hand side, there is a very small region of high performance with respect to both objectives. Only the design concepts developed by elementary CA rules can be found in this region. In the middle, there is a large region composed of both designs developed by elementary CA rules and designs generated randomly. It is also clear that in this region designs developed by CA rules dominate designs produced randomly. Finally, a medium-sized region of designs characterized by good performance with respect to horizontal displacement but rather poor performance with respect to the total weight of the structural systems. Also in the latter region, the designs developed by elementary CA rules dominate designs produced randomly.

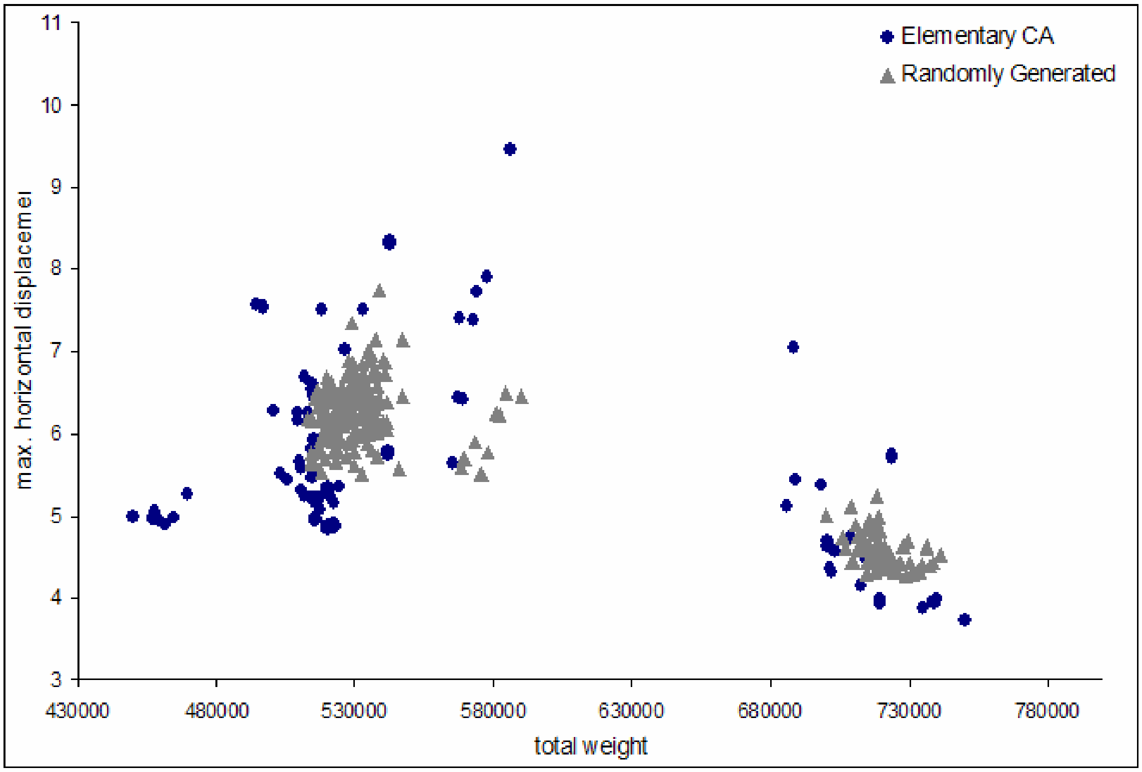
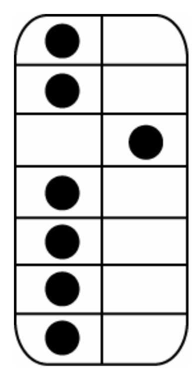


Figure 54. Comparison of design concepts with K bracings generated randomly and developed by elementary CA rules with respect to two objectives: the total weight of steel structures (horizontal axis) and their maximum horizontal displacements (vertical axis)

Design Embryo at the Top and Periodic Boundary Conditions

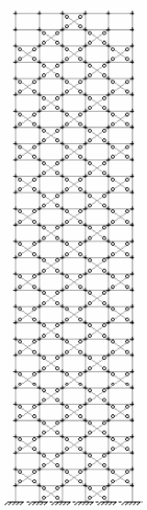
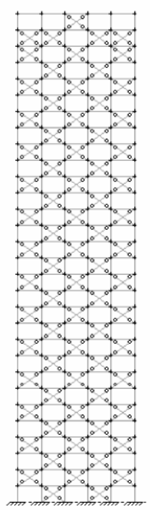
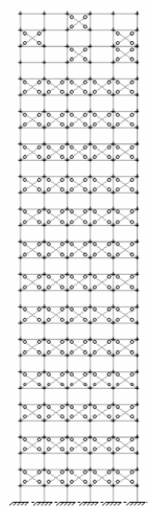
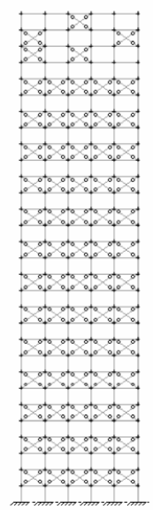
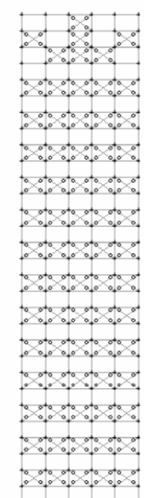
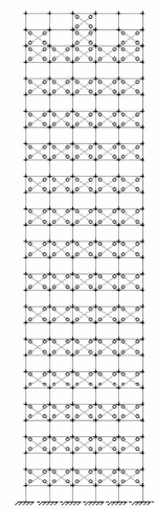
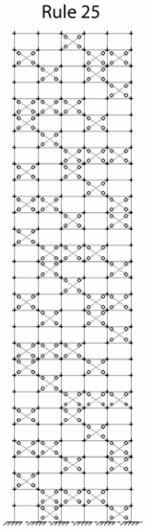
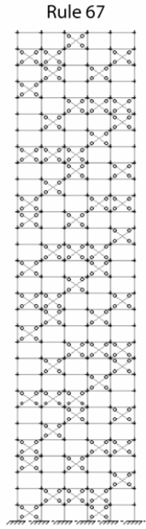
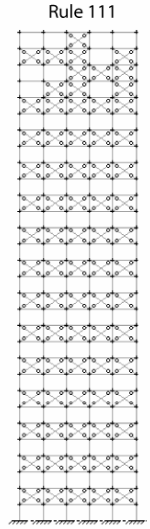
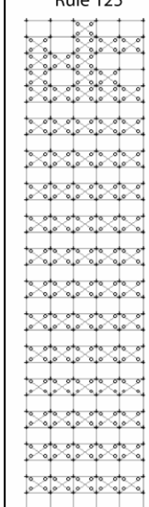
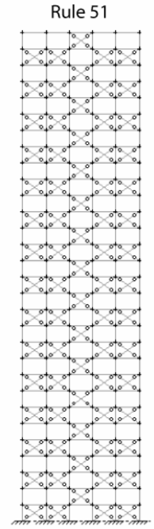
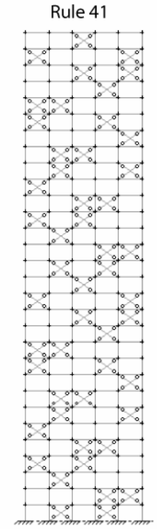


In the experiments reported in the previous section, the location of the design embryo was arbitrarily chosen at the bottom of the steel structural system. In this section, the results of the experiments are described in which the design embryo was located at the top of the structural system, and the design concepts of wind bracing systems were developed downwards. They were subsequently compared with the design concepts obtained in the experiments reported in the previous section. As earlier, the experiments were conducted for two groups of bracings elements, i.e. the group consisting of simple X bracings and no bracings (empty cells), and the other group with K bracings and no bracings.

Best Design Concepts

The 12 best design concepts (in terms of their total weight calculated using the P-Δ analysis) produced by elementary CA rules with the design embryo located at the top and with the first group of wind bracing elements (simple X bracings and no bracings) are presented in Table 18. When we compare the design concepts shown in Table 18 with the ones included in Table 8 (where the design embryo was located at the bottom), we clearly see that there are only 2 out of 12 design rules that repeat in both tables, namely rule 50 (and at the same time rule 178 which produced exactly the same design concept) and rule 51. Incidentally, the 2 repeating design rules have exactly switched order when we compare Table 18 to Table 8, i.e. rule 51 produces the best design when the design embryo is located at the bottom (Table 8) and 11th best design when the design embryo is located at the top (Table 18) whereas rule 50 (and rule 178) develops 11th best design when the design embryo is located at the bottom (Table 8), and the best design when the design embryo is located at the top (Table 18).

Table 18. Best designs of wind bracing systems consisting of simple X bracings produced by elementary CA rules with periodic boundary conditions and the design embryo located at the top

Rule 50, 178  553,912 556,223 4.4353 4.4857	Rule 179  556,951 560,292 4.4329 4.4818	Rule 9  562,044 566,843 6.6408 6.9954	Rule 65  562,044 566,843 6.6228 6.9767	Rule 37  562,962 567,238 6.6233 6.7269	Rule 109  565,814 569,926 6.5902 6.9018
Rule 25  565,933 571,086 5.2416 5.3129	Rule 67  565,933 571,086 5.2204 5.2918	Rule 111  568,597 573,052 6.1157 6.2384	Rule 125  568,597 573,052 6.0986 6.2213	Rule 51  568,678 574,538 4.9506 4.9781	Rule 41  573,429 574,628 5.4016 5.5189

The best design developed using the design embryo located at the top of the structural system (rule 50 and 178) in better than the one generated using the design embryo located at the bottom (rule 51) by about 6,350 lbs. Also, when we compare two design concepts developed using the same rule 50, or rule 178, but with different locations of the design embryo, i.e. at the bottom and at the top, we observe that when the design embryo is placed at the top, we obtain a better design concept in terms of the total weight of the structural system by about 14,000 lbs. (calculated using P-Δ analysis), or 2.4 percent. The reduced steel consumption is achieved at a cost of increasing maximum horizontal displacement of the structural system by 0.07 inch, or 1.7 percent.

Design Embryo at the Top vs. at the Bottom

However, when we compare all pairs of the design concepts developed using the same design rules but with different locations of the design embryo (top and bottom) we observe no significant differences, as it is shown in Table 19. The median total weight reduction and horizontal displacement reduction are close to 0. 138 design concepts out of all 256 design concepts (or 147 when the P-Δ analysis is conducted) developed by elementary CA rules from the design embryo located at the top, have a reduced total weight of the structural system, when compared to the design concepts developed using the same rules but from the design embryo located at the bottom. This roughly corresponds to half of the design concepts with a reduced total weight and half of the design concepts with an increased total weight.

Table 19. Comparison of the design concepts composed of simple X bracings and generated by elementary CA rules with the design embryo located at the bottom and at the top of a steel structural system

Quantity	*1st order*	*P-Δ*
Number of design concepts with reduced weight	138	147
Number of design concepts with increased horizontal displacement	85	116
Median weight reduction	0 lbs.	0 lbs.
Median percentage of weight reduction	0 %	0 %
Median displacement reduction	0.0073 in.	0.0001 in.
Median percentage of displacement reduction	0.43 %	0.01 %

K Bracings

The same experiments were repeated with the second group of bracing elements (K bracings and no bracings). The 12 best designs produced in these experiments are shown in Table 20. 11 designs shown in this table represent various variations of the fully braced frame. The 12th best design, the design developed by rule 179, is the only design that exhibits qualitatively different structural shaping pattern, i.e. the checkerboard pattern. When we compare Table 20 with Table 16, we observe that 6 out of 12 best designs shown in both tables are generated by the same rules, i.e. rule 151 (and others generating the same pattern), 222 (and 254), 251, 235, 249, and 179. In both cases, i.e. when the design embryo is located at the bottom and at the top, the best design produced by elementary cellular automata is developed by rule 151. Rule 151 develops a better design concept (in terms of the total weight of the steel structural system calculated using P-Δ analysis) when the embryo is located at the bottom. The reduction of the steel consumption is, however, almost negligible and equal to 879 lbs., or 0.2 percent.

The comparison of all pairs of design concepts developed using the same design rules but with different locations of the design embryo (top and bottom) is presented in Table 21.

Table 20. Best designs of wind bracing systems consisting of K bracings produced by elementary CA rules with periodic boundary conditions and the design embryo located at the top

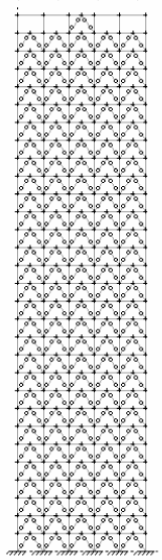
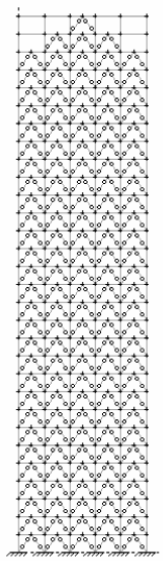
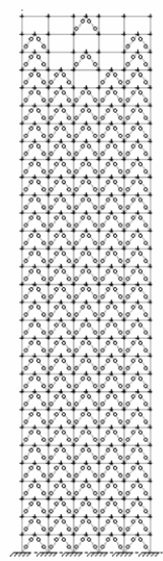
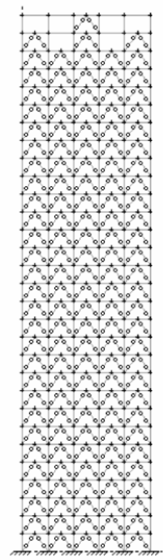
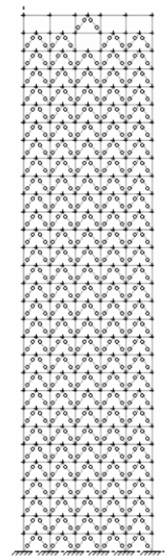
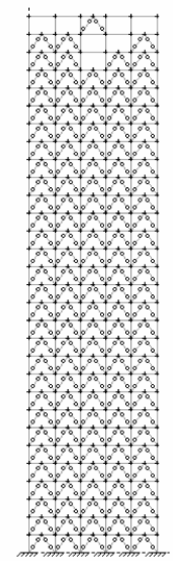
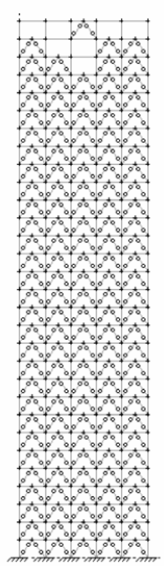
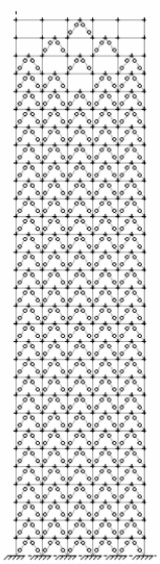
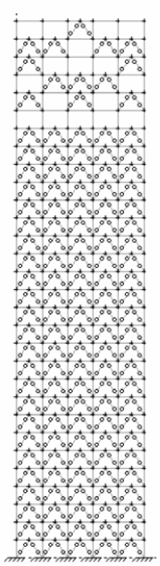
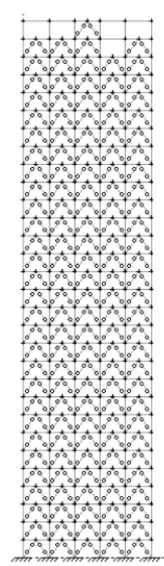
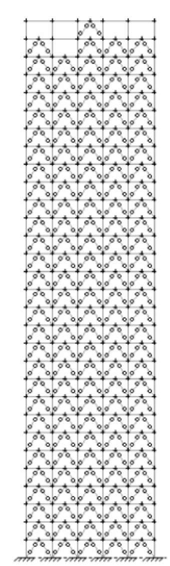
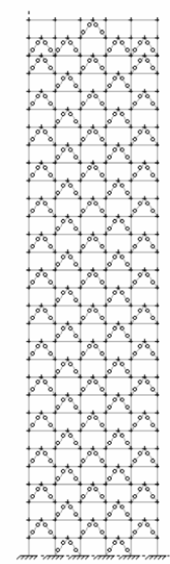
Rule 151, 159, 183, 191, 215, 223, 247, 255	Rule 222, 254	Rule 233	Rule 237	Rule 251	Rule 235
					
454,708 453,899	460,933 460,126	461,376 460,572	461,520 460,716	462,099 461,295	462,331 461,358
4.9105 5.0240	4.9092 5.0226	4.9638 5.0795	4.9230 5.0372	4.9089 5.0223	4.9288 5.0415
Rule 249	Rule 250	Rule 147	Rule 239	Rule 253	Rule 179
					
462,331 461,358	465,021 465,035	467,204 467,626	468,935 468,947	468,935 468,947	674,683 495,786
4.9371 5.0524	4.8517 4.9604	5.4256 5.5841	4.8388 4.9471	4.8469 4.9552	4.3086 5.6495

Table 21. Comparison of design concepts with K bracings generated by elementary CA rules with the design embryo located at the bottom and at the top of a steel structural system

Quantity	*1st order*	*P-Δ*
Number of design concepts with reduced weight	124	148
Number of design concepts with increased horizontal displacement	134	104
Median weight increase	1,204 lbs.	0 lbs.
Median percentage of weight increase	0.23 %	0.0 %
Median displacement reduction	-0.0003 in.	0.0016 in.
Median percentage of displacement reduction	-0.004 %	0.082 %

Table 21 shows that about half of the design concepts (124 for the first order structural analysis and 148 for P-Δ analysis) have a reduced total weight when the design embryo is located at the top. Roughly the same proportions are obtained with respect to the number of designs with an increased horizontal displacement. The median estimates for the total weight increase and the horizontal displacement reduction are close to zero, similarly as it was the case with the first group of wind bracing elements (simple X bracings and no bracings).

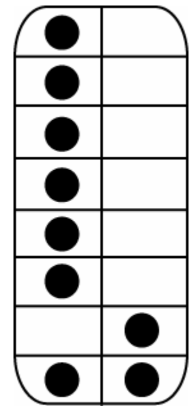
Concluding, the experimental results presented in this subsection show that the **location of the design embryo has no impact on the quality of the obtained design concepts**. Both in the case of the first group (simple X bracings and no bracings) and the second group (K bracings and no bracings) of wind bracing elements, the obtained median estimates are close to zero (see Table 19 and Table 21).

On the other hand, as it was discovered in the previous subsection (see Table 17), **there are significant differences between the design concepts composed of simple X bracings and the design concepts consisting of K bracing**s. This fact is further confirmed by Table 22 which shows that the design concepts developed from the design embryo located at the top and composed of K bracings use about 13 percent less steel than the same design concepts consisting of simple X bracings. At the same time, they exhibit about 22 percent larger horizontal displacements.

Table 22. Comparison of design concepts generated by elementary CA rules utilizing simple X bracing and K bracing elements with the design embryo located at the top

Quantity	*1st order*	*P-Δ*
Number of design concepts with reduced weight	204	172
Number of design concepts with increased horizontal displacement	195	185
Median weight reduction	103,791 lbs.	72,972 lbs.
Median percentage of weight reduction	12.9%	12.0%
Median displacement increase	0.9891 in.	0.9852 in.
Median percentage of displacement increase	23.0%	22.4%

Design Embryo at the Bottom and Nonperiodic Boundary Conditions



Design experiments reported so far involved exclusively elementary CAs with periodic boundary conditions. However, nonperiodic boundary conditions seem more natural for the design problems considered in this dissertation. For example, it is hard to imagine from a structural design point of view that a wind bracing element located in the rightmost bay strongly interacts with the wind bracing element located in the leftmost bay, and vice versa. Thus, elementary CAs with nonperiodic boundary conditions seem to be more appropriate to represent interactions among structural members in the considered design problems. The results of the design experiments investigating these types of concept generation mechanisms are reported in this subsection.

In the design experiments with elementary CAs and nonperiodic boundary conditions the following assumptions regarding the boundaries were made:

- The value of the left neighbor of the leftmost cell in the initial configuration of wind bracing elements (design embryo) was assumed to be equal to no bracing, i.e. it had value of 0 (see Figure 19).
- Similarly, the value of the right neighbor of the rightmost cell in the initial configuration of wind bracing elements was assumed to be equal to no bracing.
- The same boundary conditions (no bracing as a left neighbor for the leftmost cell and as a right neighbor for the rightmost cell) were applied to all configurations of wind bracings at subsequent stories which were obtained during the process of iteration of an elementary CA.

A graphical illustration of this process is presented in Figure 55. Even though the design embryo and the design rule are exactly the same as the ones used in Figure 26 where elementary CAs with periodic boundary conditions were investigated, the developed structural shaping pattern shown in Figure 55 is different than the one presented in Figure 26.

Similarly as in the previous two subsections, the experiments conducted with elementary CAs with nonperiodic boundary conditions used both groups of wind bracings elements. Depending on the group of wind bracing elements, the entire set of 256 elementary CA rule were applied to either the design embryo shown in Figure 50a) or the one shown in Figure 50b). The collection of 256 design concepts developed by all elementary CAs with nonperiodic boundary conditions and the first group of types of wind bracing elements, i.e. no bracings and simple X bracings, is presented in Appendix C. In these experiments, the design embryo was located at the bottom of the steel structure.

Best Designs

Table 23 shows twelve best design concepts in terms of the total weight of the steel structural system (calculated using P-Δ analysis) developed using elementary CAs with nonperiodic boundary conditions and with the design embryo located at the bottom.

It shows that the best design was generated by rule 51, similarly as it was the case with elementary CAs with periodic boundary conditions. Also, when nonperiodic boundary conditions are imposed, then rule 179 develops exactly the same structural shaping pattern as rule 51. When we compare the design concept developed by rule 179, as shown in Table 23, with the designs concept developed by the same rule and the same design embryo and presented in Table 8, we can see that we obtain a dramatically different structural shaping pattern.

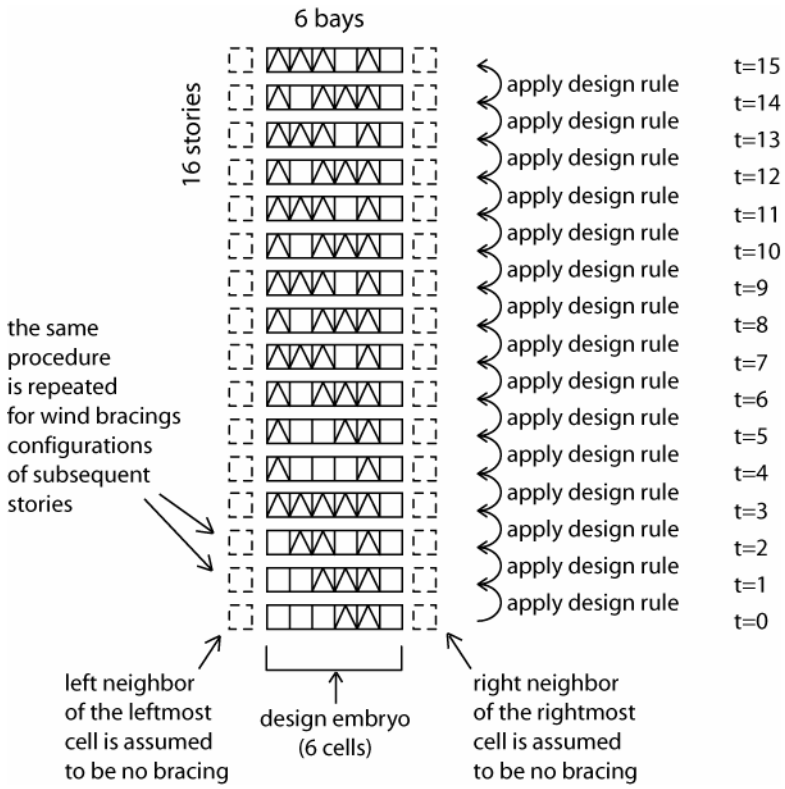


Figure 55. Graphical illustration of a process of generating a design concept of a wind bracing system using elementary CAs with nonperiodic boundary conditions

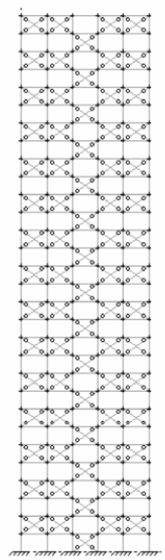
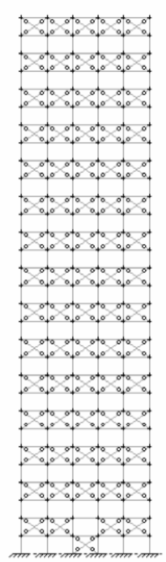
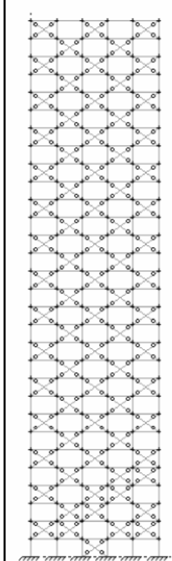
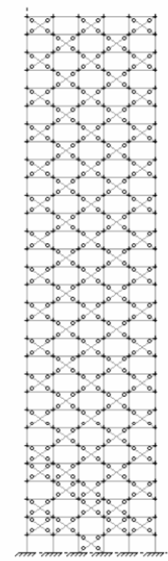
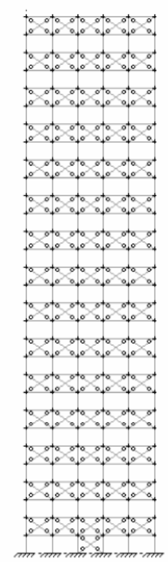
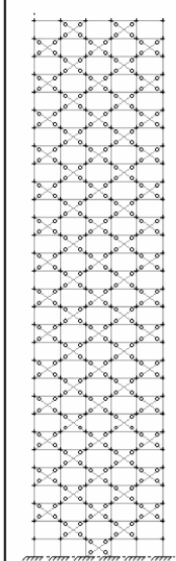
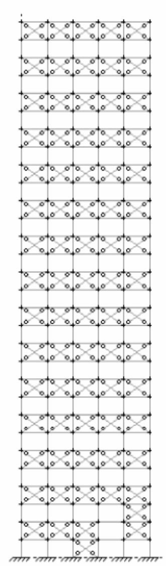
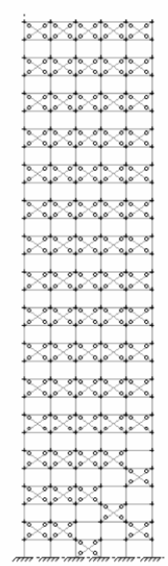
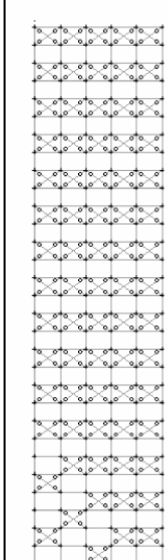
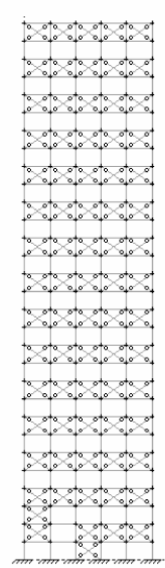
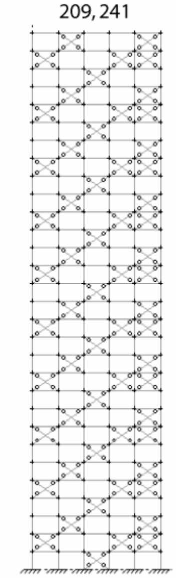
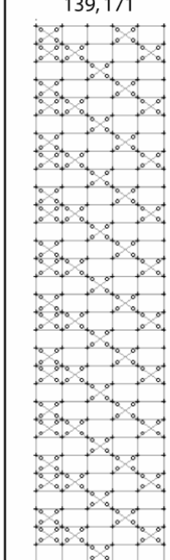
There are only 5 design concepts that appear in the group of best design generated by elementary CA rules with periodic boundary conditions and the group of designs produced with nonperiodic boundary conditions. They include the previously mentioned design concept developed by rule 51 as well as designs generated by rules 19, 50 (and 178), 23 (and 55), and 7.

Structural Shaping Patterns

Half of the design concepts in the group of 12 best designs developed with nonperiodic boundary conditions exhibit the horizontal trusses pattern. Elementary CAs with nonperiodic boundary conditions generate many more design concepts with the checkerboard pattern compared to the case when periodic boundary conditions are imposed. Table 23 shows that 3rd, 4th, and 6th best design concepts developed by a total of 12 rules exhibit this structural shaping pattern. Another interesting finding is the fact that many macro bracing patterns identified in Table 12 are locally disrupted and sometimes even completely changed by nonperiodic boundary conditions. The only two examples of macro bracings patterns found among the best 12 designs in Table 23 are two mirror design concepts (11th and 12th best) that exhibit the macro bracing pattern of width equal to 1 bracing which is locally disrupted at the boundaries.

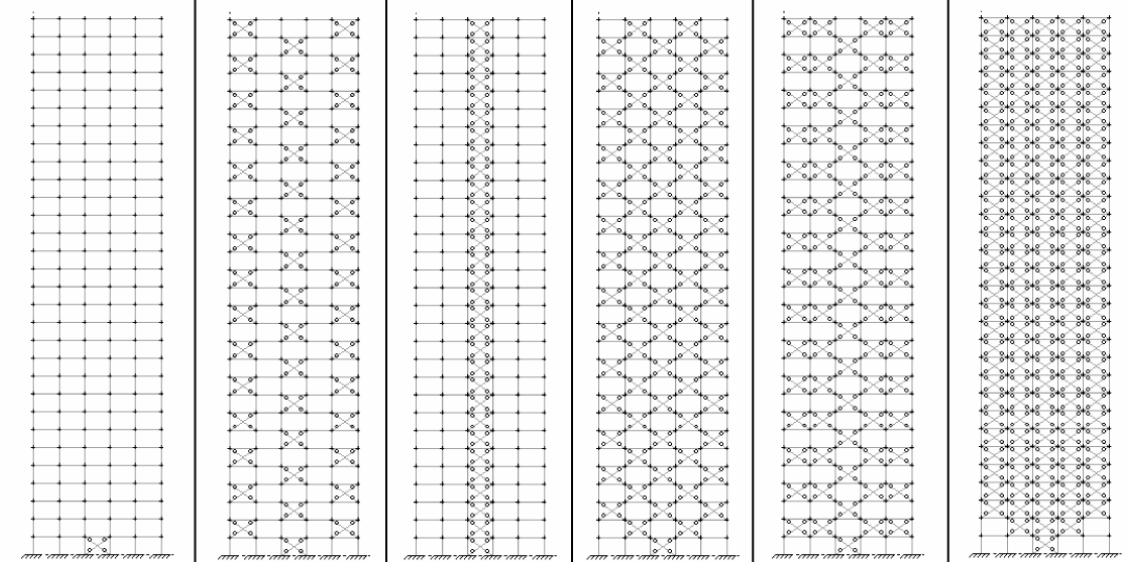
In general, when we compare the structural shaping patterns generated by the same elementary CA rules with imposed nonperiodic boundary conditions (see Appendix C) to the ones with periodic boundary conditions (see Appendix B), we can divide the developed design concepts into the following 3 groups:

Table 23. Best designs of wind bracing systems consisting of simple X bracings produced by elementary CAs with nonperiodic boundary conditions and the design embryo located at the bottom

Rule 51, 179	Rule 19	Rule 115, 243	Rule 59, 187	Rule 23, 55	Rule 50, 58, 114, 122, 178, 186, 242, 250
					
560,646 562,570 4.9940 5.1378	565,214 567,494 6.6620 6.8912	567,340 567,798 4.3417 4.4159	567,340 567,798 4.3478 4.4206	565,385 569,840 6.6824 6.8606	568,951 570,251 4.3405 4.4106
Rule 7	Rule 3, 35	Rule 17, 49	Rule 21	Rule 81, 113, 209, 241	Rule 11, 43, 139, 171
					
568,214 570,327 6.6452 6.8764	566,893 570,557 6.6264 6.8799	566,893 570,557 6.6392 6.8913	568,214 570,887 6.6247 6.8570	572,094 571,063 4.8658 4.9824	572,094 571,063 4.9224 5.0398

1. Design concepts in both cases are identical.
   Examples of elementary CA rules that generate the same design concepts with periodic and nonperiodic boundary conditions include rules 0, 1, 12, 18, 19, 23, 50, 51, etc. There are 55 designs in the entire set of 256 designs that are identical in both cases. Table 24 shows several designs concepts belonging to this group.

Table 24. Examples of design concepts which are identical no matter if periodic or nonperiodic boundary conditions are used



Rule 0	Rule 1	Rule 4	Rule 50	Rule 51	Rule 222
4,725,662 4,725,662 1.0421 1.0472	611,109 611,575 9.1900 9.3105	600,546 5,927,236 10.938 0.8975	568,951 570,251 4.3405 4.4106	560,646 562,570 4.9940 5.1378	639,149 639,399 3.9634 4.0298

2. Design concepts developed with nonperiodic boundary conditions have some local disruptions of the pattern near the structural system's boundaries.
   The disruption of the pattern can be localized within a limited space of the structural system or they can persist throughout the entire boundary region. Table 25 shows several examples of design concepts where the disruptions of the pattern are restricted to a small region close to the boundary, or close to both boundaries. Such design concepts are generated for example by rules 7, 37, and 203.
   In several instances, the local disruptions of the pattern are propagated throughout the entire boundary region of the structural system. Table 26 presents 3 examples of design rules that can be included in this group. They include rules 5, 31, 47, and others.
3. Design concepts developed with nonperiodic boundary conditions exhibit completely different pattern than the ones generated with periodic boundary conditions.
   In several cases, the local disruption of the pattern is propagated beyond the boundary region and produces a qualitatively different structural shaping pattern. Table 27 shows several examples of design concepts which exhibit completely different structural shaping patterns depending on the type of boundary conditions used.

But what is the impact of the nonperiodic boundary conditions on the performance, i.e. the total weight and the maximum horizontal displacement, of the steel structural systems?

Table 25. Examples of design concepts with some small and local disruptions of the pattern when nonperiodic boundary conditions are used

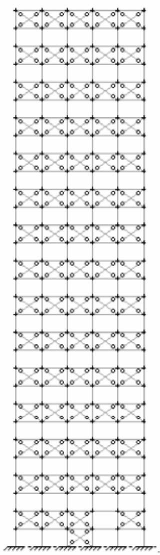
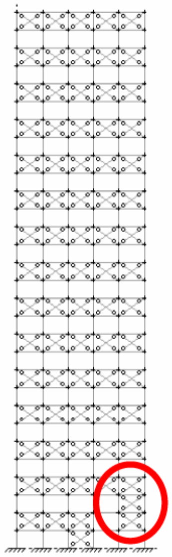
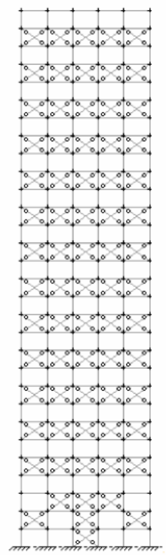
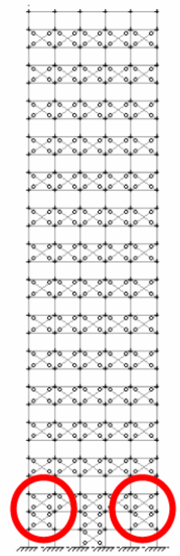
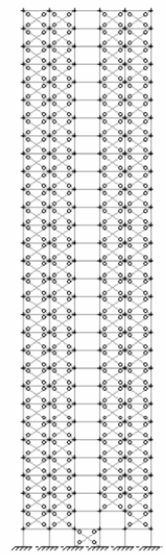
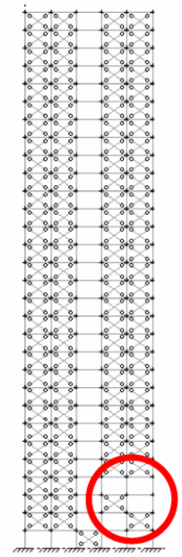
Rule 7 Periodic	Rule 7 Nonperiodic	Rule 37 Periodic	Rule 37 Nonperiodic	Rule 203 Periodic	Rule 203 Nonperiodic
					
567,943 570,439 6.6236 6.8527	568,214 570,327 6.6452 6.8764	573,463 576,834 6.2483 6.4474	577,607 580,977 6.2116 6.4089	662,833 9,490,519 7.0193 0.7015	658,273 9,366,218 6.9959 0.7254

Table 26. Examples of design concepts with local disruptions of the pattern propagating throughout the boundary region when nonperiodic boundary conditions are used

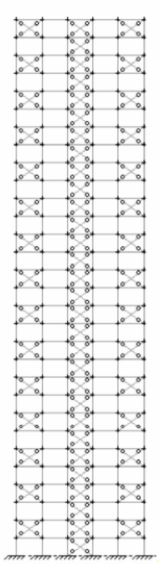
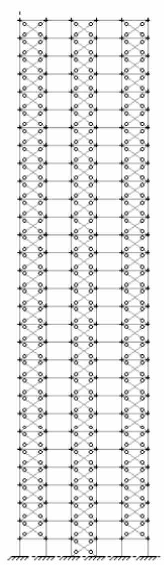
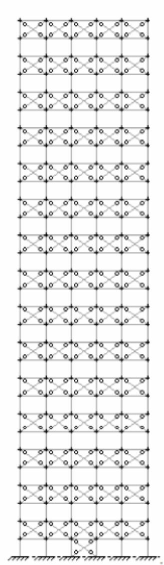
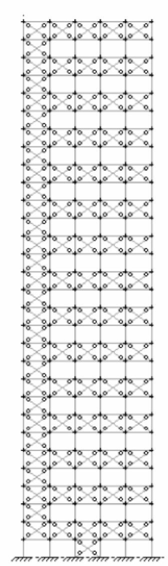
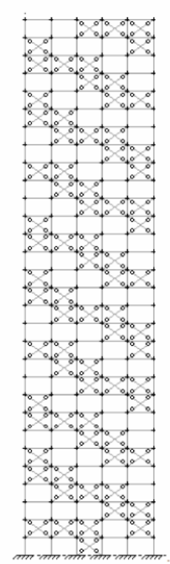
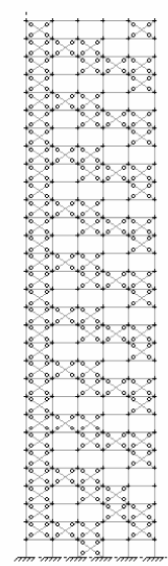
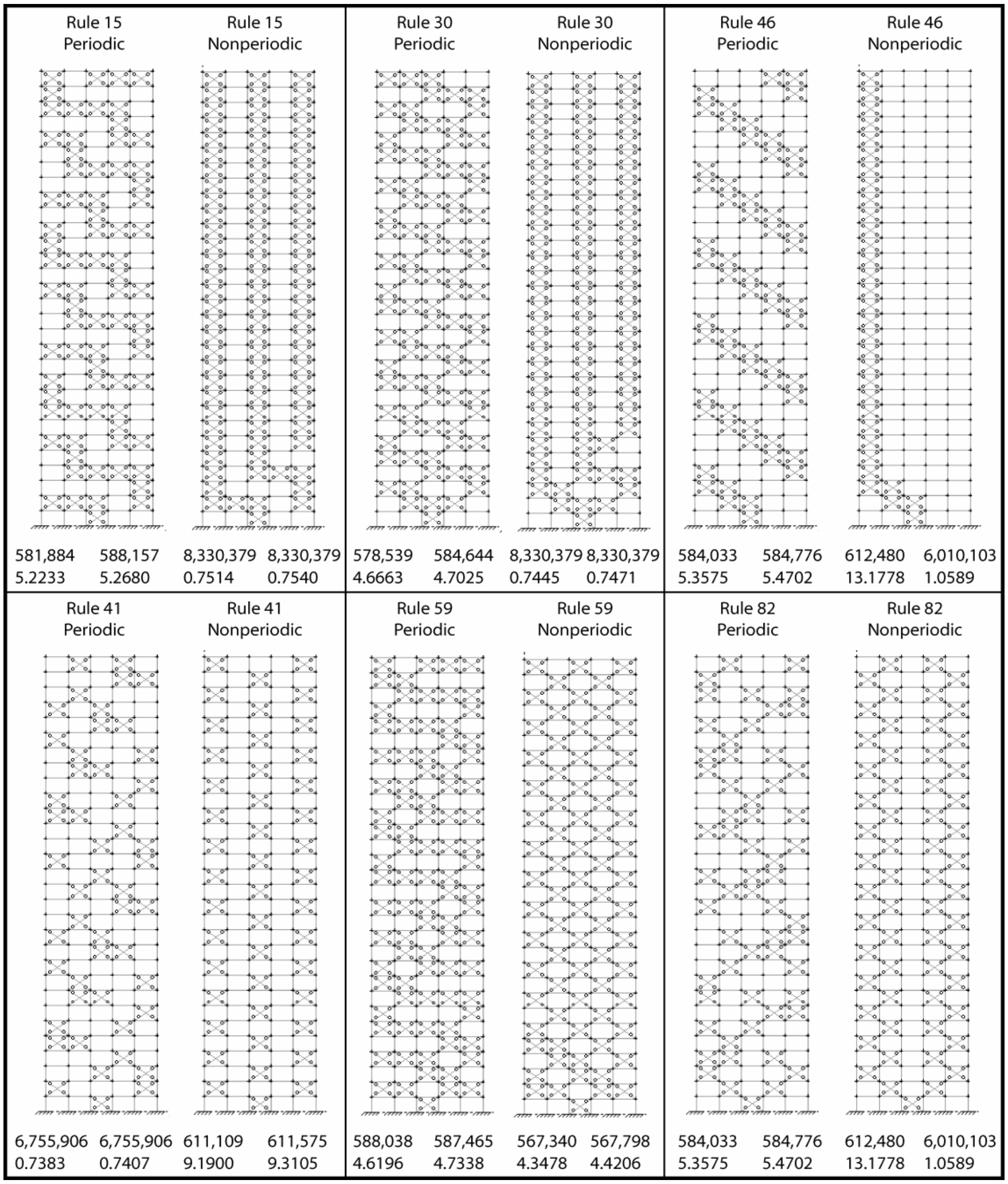
Rule 5 Periodic	Rule 5 Nonperiodic	Rule 31 Periodic	Rule 31 Nonperiodic	Rule 47 Periodic	Rule 47 Nonperiodic
					
7,170,241 628,373 0.7955 9.3504	8,330,379 8,330,379 0.7645 0.7672	565,385 569,840 6.6824 6.8606	599,846 602,841 6.0157 6.1641	594,823 597,958 5.0063 5.1034	619,478 619,661 4.9025 5.0180

Table 27. Examples of design rules generating completely different patterns when periodic and nonperiodic boundary conditions are used



Periodic Boundary Conditions vs. Nonperiodic Boundary Conditions

The best design concept developed by elementary CA rules with nonperiodic boundary conditions, i.e. the design concept generated by rule 51, is exactly the same as the one produced with periodic boundary conditions. When we compare the corresponding pairs of design concepts presented in Table 8 (periodic boundary conditions) and Table 23 (nonperiodic boundary conditions), i.e. 2nd best in Table 8 with 2nd best in Table 23, etc., we conclude that all design generated with periodic boundary conditions are better in terms of the total weight of the steel structural system (calculated using the P-Δ analysis) than the ones developed with nonperiodic boundary conditions.

However, when we take into consideration all 256 pairs of design concepts and calculate basic statistics, we observe no significant differences between the two samples Table 28 shows that when nonperiodic boundary conditions are imposed then 100 out of 256 design concepts have reduced total weight and 99 design concepts have increased total weight (55 designs are exactly the same as discussed earlier). The median reduction of the total weight and median reduction of the horizontal displacement are either very close to or even equal to 0. The negative sign at the median displacement reduction value calculated using the first-order analysis means that the horizontal displacement was increased rather than reduced. Thus, **there are no significant advantages for using nonperiodic boundary conditions over periodic boundary conditions** for the design problem considered in the experiments.

Table 28. Comparison of design concepts with simple X bracings generated by elementary CA rules with nonperiodic and periodic boundary conditions

Quantity	*1st order*	*P-Δ*
Number of design concepts with reduced weight	100	100
Number of design concepts with reduced horizontal displacement	65	91
Median weight reduction	0 lbs.	0 lbs.
Median percentage of weight reduction	0 %	0 %
Median displacement reduction	-0.0218 in.	0.0000 in.
Median percentage of displacement reduction	0.50 %	0.00 %

K Bracings

Similar experimental work has been conducted with the second group of wind bracing elements, i.e. no bracings and K bracings. Table 29 shows the 12 best designs produced in these experiments. Contrary to the experiments with the first group of wind bracings, the majority of elementary CA rules that produced best designs with nonperiodic boundary conditions also proved to perform well when periodic boundary conditions were imposed (compare designs in Table 16 with the ones in Table 29). Major differences between the contents of Table 29 and Table 16 include significantly larger number of design concepts exhibiting the checkerboard pattern produced by nonperiodic boundary conditions. Also, the design concept produced by rule 95 (and 127) has been improved (by reducing the total weight and horizontal displacements) by imposing nonperiodic boundary conditions which change the pattern along the structure's boundaries.

Table 30 presents some basic statistical estimates comparing two samples of 256 design concepts with K bracings developed using nonperiodic or periodic boundary conditions. The table shows that there are some differences when using nonperiodic boundary conditions compared to the case when periodic boundary conditions are imposed. They can be easier identified when the P-Δ analysis is conducted. Elementary CAs with nonperiodic boundary conditions generate design concepts with an increased total weight (by about 4%) and at the same with a reduced horizontal displacement (by about 1%).

Table 29. Best designs of wind bracing systems consisting of K bracings produced by elementary CAs with nonperiodic boundary conditions and the design embryo located at the bottom

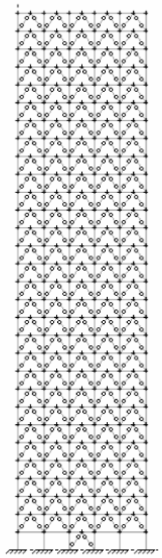
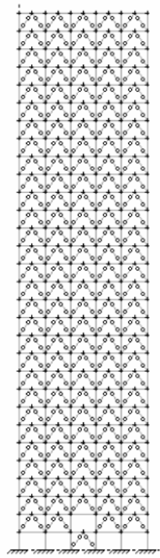
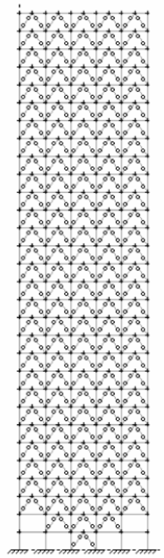
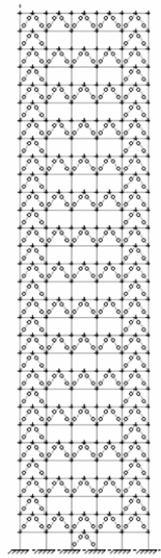
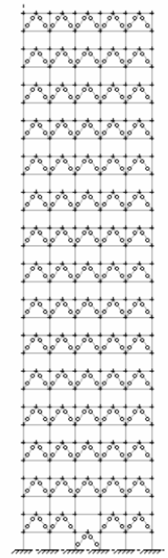
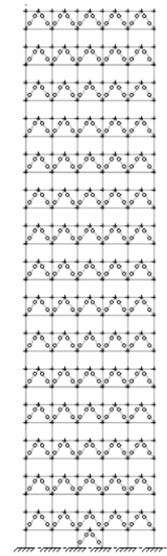
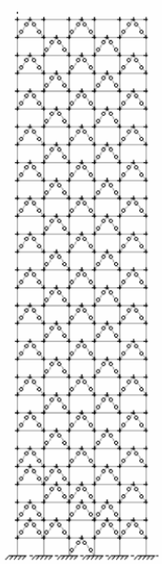
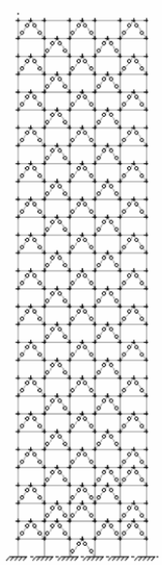
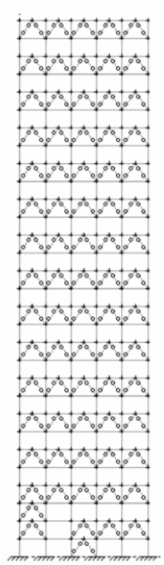
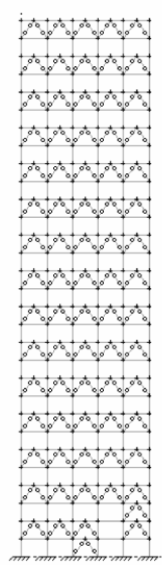
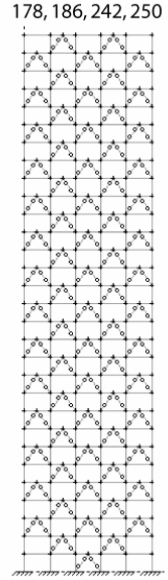
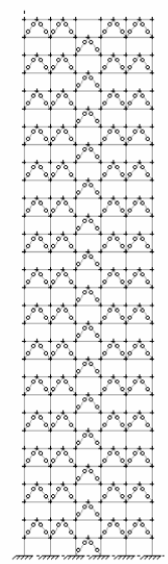
Rule 223, 255	Rule 251	Rule 222, 254	Rule 95, 127	Rule 19	Rule 23, 55
					
450,234 453,021 4.9839 5.0753	457,502 459,519 4.9571 5.0523	459,513 461,960 4.9301 5.0177	496,904 497,362 6.3700 6.5372	496,824 498,121 7.5569 7.7673	494,487 498,248 7.5732 7.7211
Rule 59, 187	Rule 115, 243	Rule 21	Rule 7	Rule 50, 58, 114, 122, 178, 186, 242, 250	Rule 51, 179
					
497,265 498,637 5.4928 5.5825	497,265 498,638 5.5187 5.6089	689,505 501,041 5.2722 7.6600	689,505 501,041 5.3062 7.6802	501,539 502,854 5.5784 5.6506	501,044 503,376 6.2758 6.4272

Table 30. Comparison of design concepts with K bracings generated by elementary CA rules with nonperiodic and periodic boundary conditions

Quantity	*1st order*	*P-Δ*
Number of design concepts with reduced weight	73	48
Number of design concepts with reduced horizontal displacement	112	126
Median weight reduction	-53 lbs.	-21,195 lbs.
Median percentage of weight reduction	0 %	-4.14 %
Median displacement reduction	0.0000 in.	0.0123 in.
Median percentage of displacement reduction	0.0000 %	1.31 %

Summary

In the last three subsections I investigated the simplest generative representations studied in this dissertation and composed of the design rule based on an elementary CA rule and the design embryo consisting of a single simple X bracing, or K bracing, located in the central bay. Even these extremely simple generative representations produced novel structural shaping patterns with good performance. I also classified the obtained structural shaping patterns with respect to the dynamical behavior CA rules that generated these patterns. In this way, four classes of the shaping patterns were identified: fixed-point behavior, periodic behavior, apparently 'chaotic' behavior, and localized propagating structures.

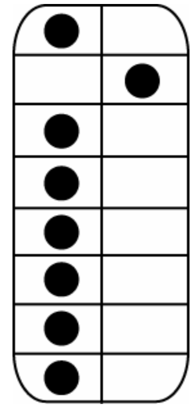
I compared the design concepts of wind bracing systems with the design concepts generated randomly and found out that they proved to perform better in terms of both the total weight of the steel structural system and its maximum horizontal displacement. The developed design concepts were also compared to designs known from the structural engineering literature. I also discovered that many traditionally known designs were generated by the simple generative representations based on elementary CAs.

In the conducted experiments, I compared two groups of wind bracing elements. The first group consisted of simple X bracings and no bracings (empty cells) while the second group included K bracings and no bracings. The results of the experiments have shown that the same structural shaping patterns consisting of simple X bracings were heavier than the ones composed of K bracings. On the other hand, simple X bracings produced structural designs with significantly better stiffness (smaller horizontal displacements).

The results of the design experiments investigating the impact of the representation parameters have shown that there is no difference in terms of performance of generated designs when the design embryo is located at the top. Hence, in the remainder of this dissertation, I will assume the location of the design embryo at the bottom of a structural system.

Another set of design experiments has shown that there is no benefit in using nonperiodic boundary conditions when simple X bracings are used. On the contrary, they may increase the total weight of the steel structural systems by several percent when K bracings are used. Their usage can be justified only when better stiffness performance of wind bracing systems consisting of K bracings is desired. In this case, however, usually simple X bracings are preferred because they provide better stiffness of a steel structural system as it was shown earlier in Table 17. Thus, periodic boundary conditions have been assumed in further experiments reported in this dissertation.

6.2.2. Randomly Generated Design Embryo



So far, I have investigated the simplest design rules based on elementary CAs and the simplest design embryos consisting of a single simple X bracing (or a K bracing) in the middle of the initial configuration of cells.

In this section, I will consider slightly more complicated experimental settings in which the design embryos are no longer restricted to assume the simplest configuration described above but may take on any possible configuration of wind bracing elements. In fact, in the experiments reported in this section, the configurations of design embryos were generated randomly. The design rules investigated here were exactly the same as before, i.e. based on elementary CA rules.

The design experiments involved the entire set of 256 elementary CA rules. Each rule was applied to 5 randomly generated design embryos and developed 5 design concepts of wind bracing systems from them. As before, the experiments were conducted with two groups of wind bracing elements, i.e. group No.1 consisting of no bracing and simple X bracing and group No.2 containing K bracings and no bracings. This time, however, each developed design concept was evaluated using the first-order analysis only. The values of the total weight of the steel structural system and its maximum horizontal displacement were recorded.

Best Designs

The best design concepts (in terms of the total weight of the steel structural system) produced by the group No.1 are presented in Table 31. The overall best design concept found in this group of experiments was generated by rule 154. Its total weight was equal to 550,336 lbs. When we compare the total weight of this design with the total weight of the best design generated from the simple design embryo (see Table 8), we observe that the weight was reduced by more 10,000 lbs. In fact, 8 out of 12 best designs shown in Table 31 have better performance, i.e. smaller total weight, than the best design generated by rule 51 from a simple design embryo.

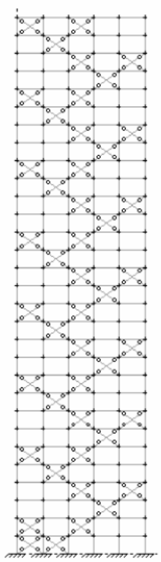
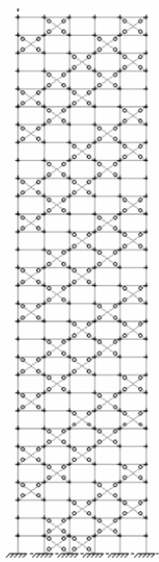
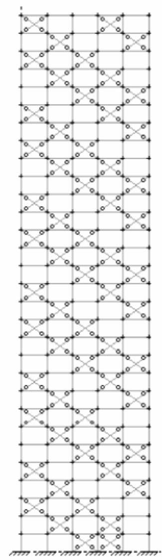
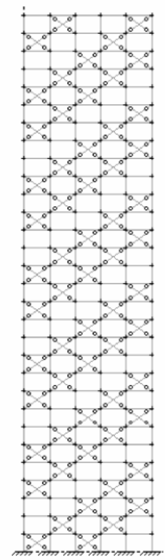
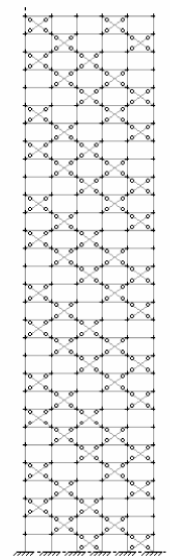
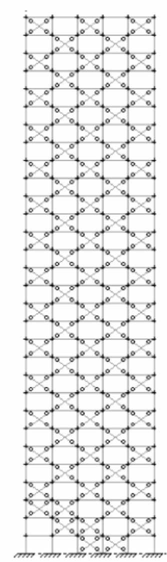
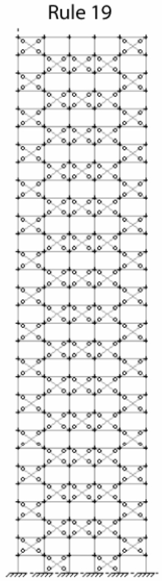
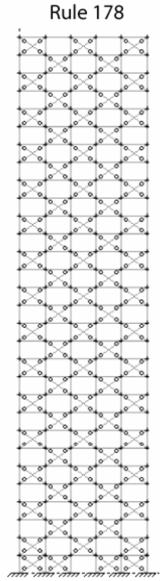
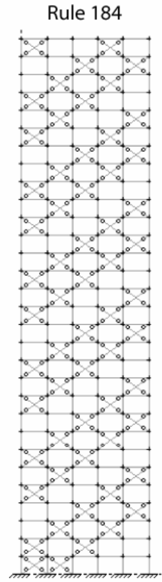
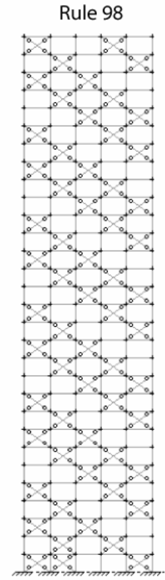
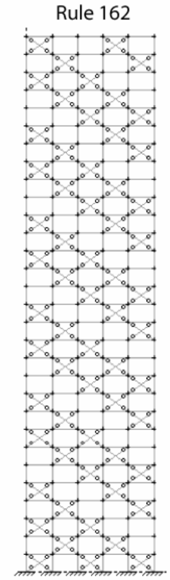
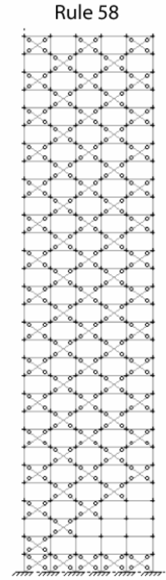
Structural Shaping Patterns

Design concepts shown in Table 31 exhibit all four major types of structural shaping patterns. The most common pattern occurring in this group is the previously identified pattern of macro bracings of width 2 in which the simple X bracings are located at a distance equal to 1 story. 7 out of 12 best designs found in Table 31 exhibit several variations of this pattern. The checkerboard pattern occurs in 3 design concepts shown in Table 31, including the concepts generated by rules 186, 178, and 58.

There are also two novel structural shaping patterns of very good performance in the group of design concepts shown in Table 31. First, rule 154 generated the best design concept with an intriguing ‘tree like’ pattern. Second, rule 19 developed an interesting pattern in which horizontal trusses located every two stories are connected through simple X bracings situated in the outer bays of the intermediate stories.

Examples of several other interesting structural shaping patterns produced by elementary CA rules with randomly generated design embryos are presented in Table 32. The group of design concepts shown here includes diverse structural patterns of reasonable performance. It is worth mentioning that even greater diversity of generated structural shaping patterns was obtained by relaxing the assumption of using only the simplest design embryo.

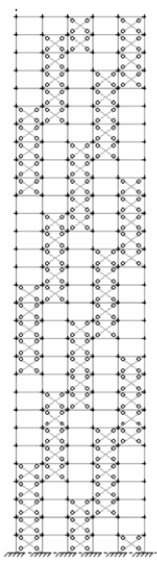
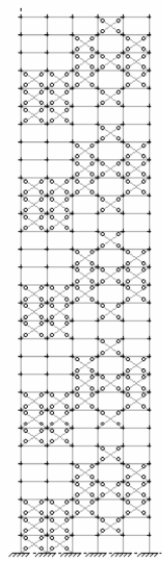
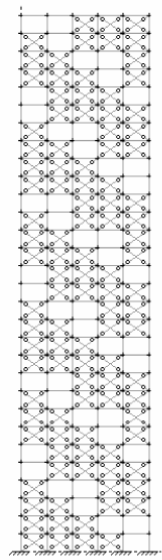
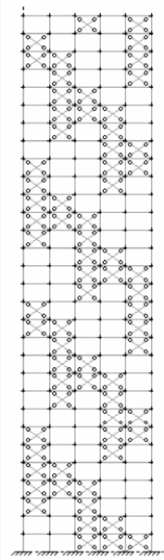
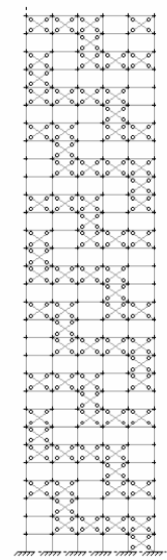
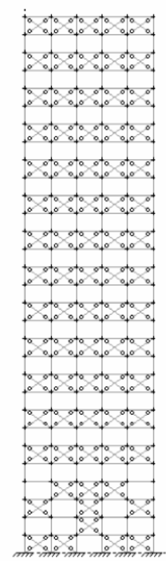
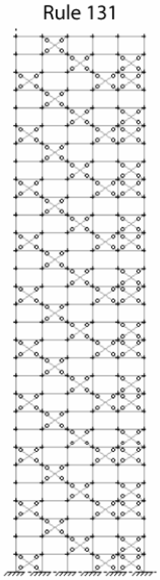
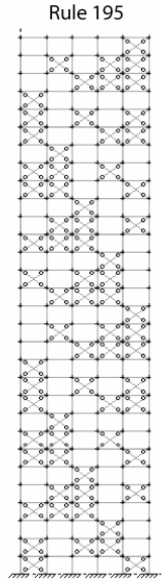
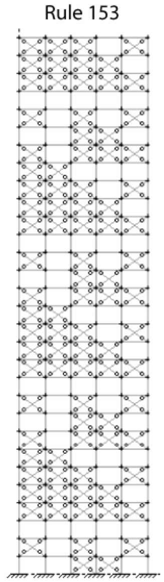
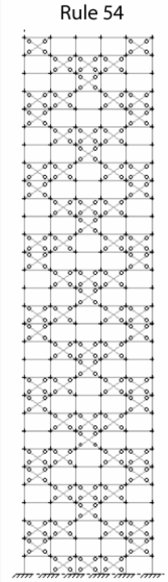
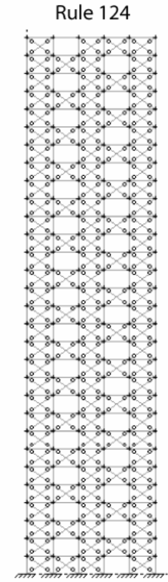
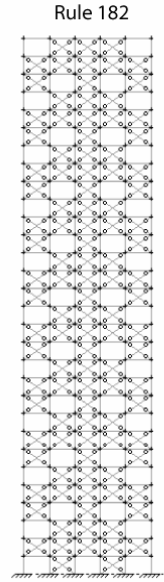
Table 31. Best designs of wind bracing systems consisting of X bracings produced by elementary CAs with a randomly generated design embryo located at the bottom

Rule 154  550,366 5.6417	Rule 184  558,464 4.6932	Rule 98  558,464 4.7601	Rule 48, 241, 248  559,229 4.6979	Rule 162, 170, 234  559,229 4.7648	Rule 186  559,381 4.4702
Rule 19  559,892 5.3018	Rule 178  559,935 4.4465	Rule 184  560,673 4.8126	Rule 98  560,824 4.7197	Rule 162  560,824 4.7283	Rule 58  561,898 4.5247

K Bracings

Another set of experiments with the second group of wind bracing elements also showed that elementary CA rules applied to randomly generated design embryos produce superior results. The best design concepts generated in these experiments are presented in Table 33. The total weight of the best design concept was equal to 449,521 lbs. It was the best design generated by elementary CA rules. In fact, all 12 best design concepts shown in Table 33 outperformed the best design concepts generated by elementary CA rules with an arbitrarily assumed design embryo.

Table 32. Interesting structural shaping patterns produced by elementary CA rules with randomly generated design embryos

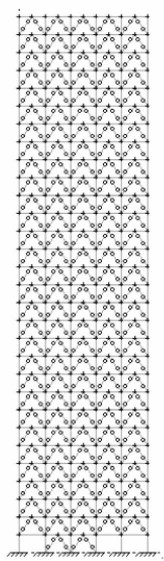
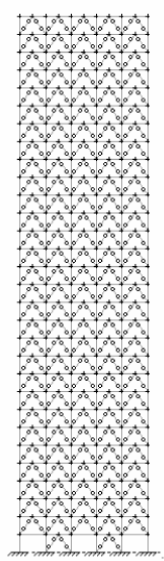
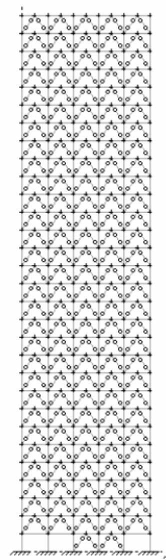
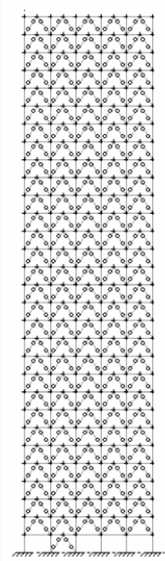
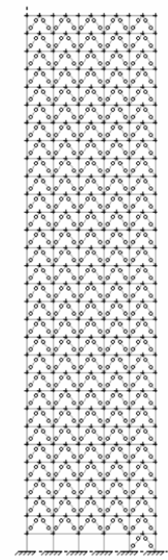
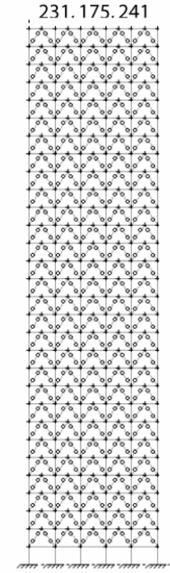
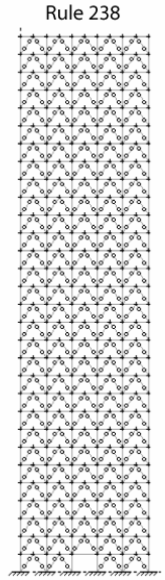
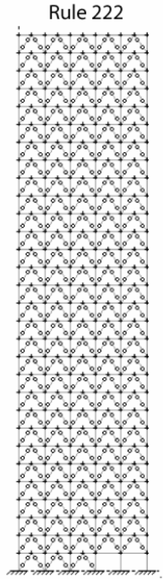
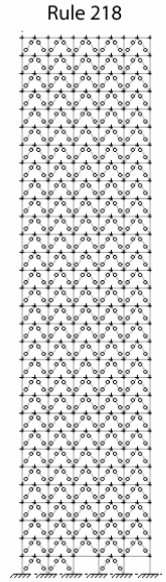
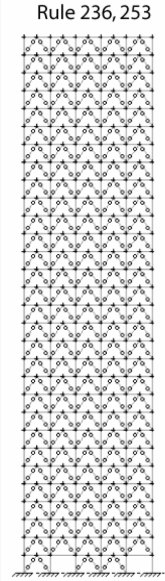
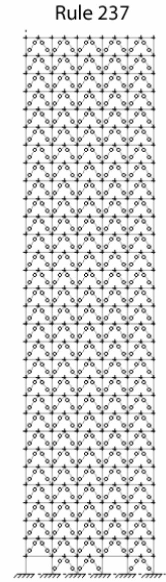
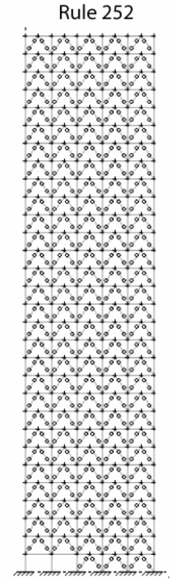
Rule 85	Rule 105	Rule 155	Rule 173	Rule 15	Rule 37
					
570,667 4.8043	620,017 4.5864	608,037 4.2896	595,751 4.7950	584,808 5.2439	569,500 6.6639
Rule 131	Rule 195	Rule 153	Rule 54	Rule 124	Rule 182
					
581,961 5.1101	587,991 4.8807	610,906 5.1980	569,940 4.6831	637,123 4.1947	603,084 4.1574

Structural Shaping Patterns

All design concepts presented in Table 33 exhibit the fully braced pattern in which all cells from the second story up to the topmost story are occupied by K bracings. The differences among the design concepts are restricted to various configurations of the design embryo.

Interestingly, the fully braced pattern in which the entire configuration of a wind bracing system, including all cells of the design embryo, is composed of K bracings turned out not to be the optimal design concept with respect to the total weight of the structural system. Its total weight was more than 8,500 lbs. larger than the best design concepts shown in Table 33.

Table 33. Best designs of wind bracing systems composed of K bracings produced by elementary CA rules with a randomly generated design embryo located at the bottom

Rule 255	Rule 184	Rule 255	Rule 183, 223	Rule 183	Rule 131, 141, 177, 231. 175. 241
					
449,521 4.9813	449,521 4.9844	449,521 4.9897	449,776 4.9883	449,776 5.0049	449,891 5.0111
Rule 238	Rule 222	Rule 218	Rule 236, 253	Rule 237	Rule 252
					
450,096 4.9893	450,209 4.9613	450,209 4.9613	450,209 4.9639	450,209 4.9673	450,209 4.9765

The design experiments with randomly generated design embryos and the second group of wind bracing elements also generated interesting structural shaping patterns of good performance. Several examples of such patterns are presented in Table 34. Similarly as before, when elementary CA rules are applied to randomly generated design embryos, they produce even greater diversity of interesting structural shaping patterns.

Table 34. Interesting structural shaping patterns composed of K bracings and produced by elementary CA rules with randomly generated design embryos located at the bottom

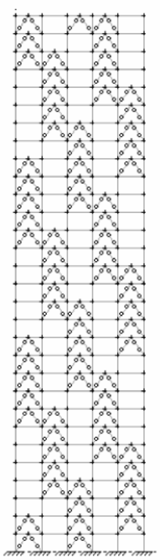
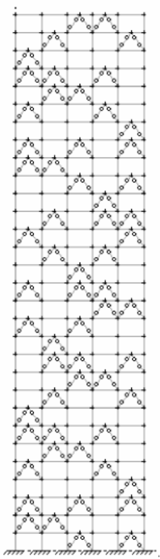
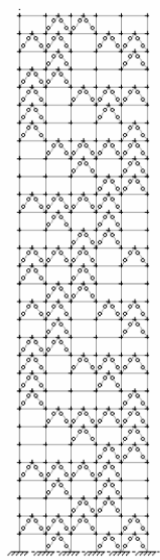
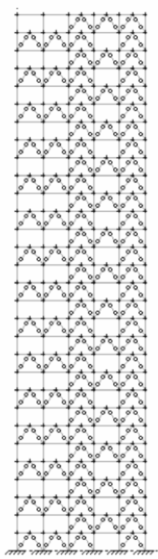
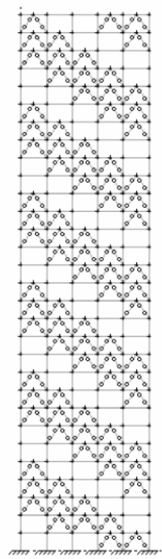
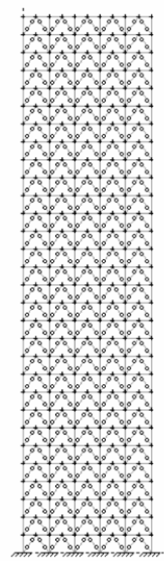
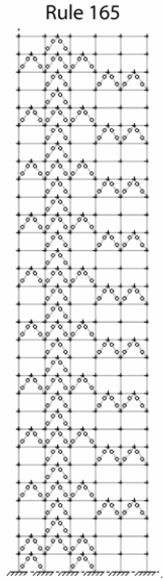
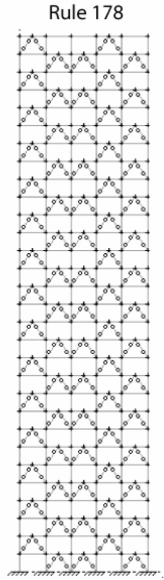
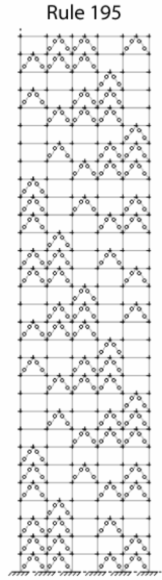
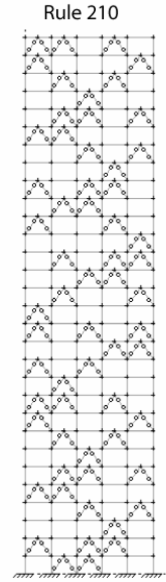
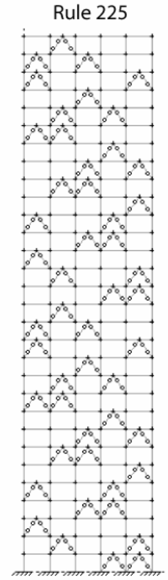
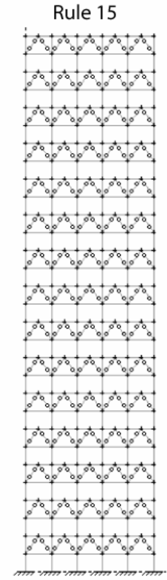
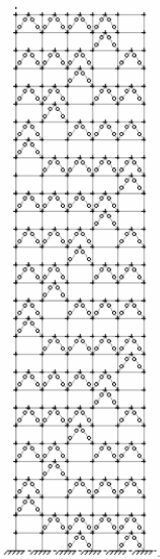
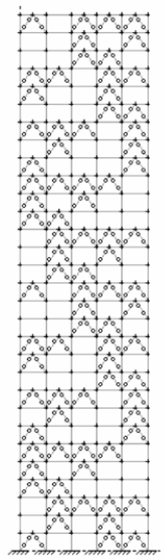
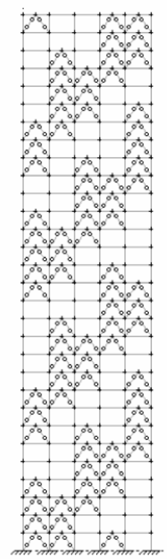
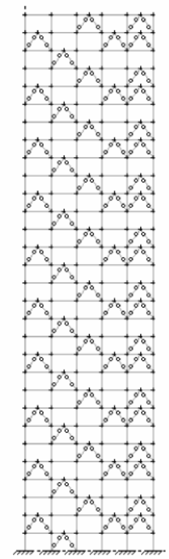
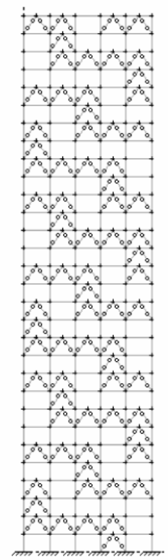
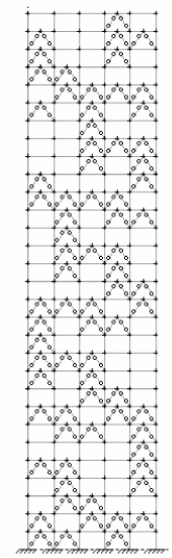
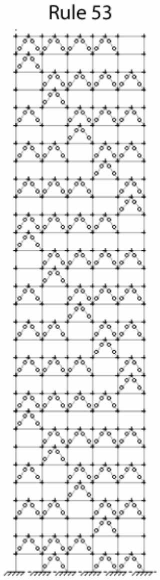
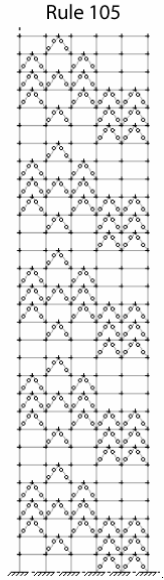
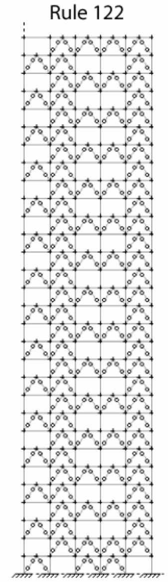
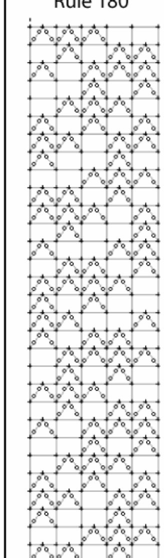
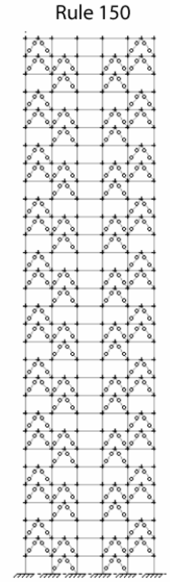
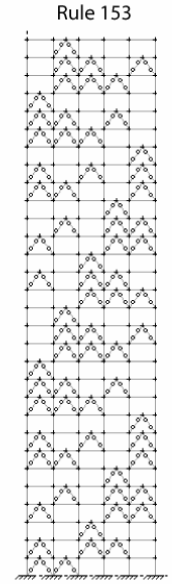
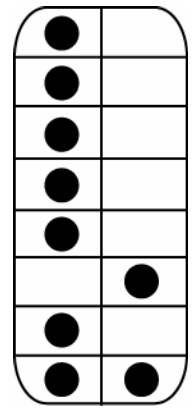
Rule 14	Rule 26	Rule 102	Rule 122	Rule 183	Rule 150
					
514,347 5.7634	512,206 6.5097	517,161 5.6908	535,228 5.7854	523,861 5.4125	458,274 4.8721
Rule 165	Rule 178	Rule 195	Rule 210	Rule 225	Rule 15
					
539,623 7.0209	508,944 5.7628	524,994 6.1681	508,576 6.4537	528,823 6.9707	494,080 7.5761

Table 34 cont. Interesting structural shaping patterns composed of K bracings and produced by elementary CA rules with randomly generated design embryos located at the bottom

Rule 39	Rule 60	Rule 88	Rule 131	Rule 15	Rule 45
					
508,567 6.1639	518,273 5.9086	523,699 5.7094	516,684 6.6542	534,981 6.1193	526,450 5.9097
Rule 53	Rule 105	Rule 122	Rule 180	Rule 150	Rule 153
					
508,124 6.2887	540,035 5.9496	524,456 5.6086	516,649 5.3354	551,117 7.7246	522,161 6.2404

Thus, the design experiments with both groups of wind bracing elements have shown that by making the generative representation based on elementary CA slightly more general (and less constrained), not only the qualitative difference has been achieved (larger selection of structural patterns) but also the performance of generated design concepts has been improved. This fact will be utilized in the morphogenic design experiments reported in chapter 8 in the following way. Instead of using fixed design embryos and evolving only design rules which are subsequently applied to the design embryos both the embryo and the rule will be evolved. The optimal configurations of both elements of the generative representation, i.e. the embryo and the rule, will be sought.

6.2.3. Design Concept Generators with Symmetry Constraint



In this section, I will demonstrate how we can make use of background knowledge on the design problem and incorporate it in the generative representation. The domain knowledge effectively reduces the size of the design space and acts as a constraint. I will illustrate that with a simple example of a symmetry constraint. In this case, I will apply the structural design knowledge that symmetric structural systems usually outperform asymmetric ones. Next, I will show how we can incorporate this knowledge in the generative representation of a wind bracing system in order to develop symmetric design concepts. Furthermore, I will demonstrate that elementary CA rules with an imposed symmetry constraint develop symmetric design concepts from the symmetric design embryos. I will also compare the symmetric design concepts with asymmetric ones in terms of their total weight and their maximum horizontal displacements.

Symmetry of structures is an important property from a structural engineering perspective. Almost all steel structural systems are symmetric and that is considered highly desirable for various reasons (aesthetics, constructability, structural behavior, etc.). Thus, symmetry is one of the most frequently used requirements in structural design.

The process of imposing a symmetry constraint on the design rules based on elementary CA rules is straightforward and consists of two steps:

1. Imposing a so-called reflection symmetry on CA rules (Wolfram 1983), and
2. Imposing symmetry on the design embryos.

In the case of elementary CA rules, the reflection symmetry introduces two constraints on the CA rule. First, the CA rule has to yield the same outcome value for the local neighborhoods 100 and 001. Second, the CA rule must give the same outcome value for the local neighborhoods 110 and 011. Graphical illustration of the reflection symmetry property is presented in Figure 56.

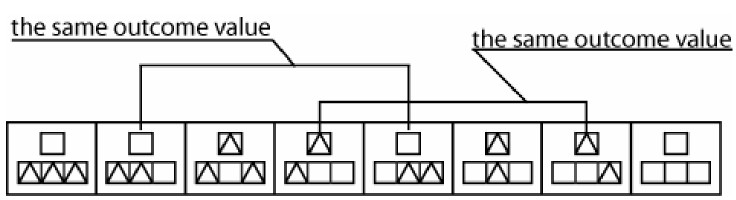


Figure 56. Graphical illustration of the reflection symmetry of a design rule based on an elementary CA rule

When the reflection symmetry is imposed, the space of design rules based on elementary CA rules is restricted to 64 rules (compared to 256 rules when no constraint is imposed) of the form:

$$\alpha_1\alpha_2\alpha_3\alpha_4\alpha_2\alpha_5\alpha_4\alpha_6,$$

where $\alpha_i \in \{0,1\}$ and the same ordering of the local neighborhoods is assumed as in Figure 56.

Symmetry of the design embryo is another necessary condition in order to produce symmetric design concepts. Even symmetric design rules do not necessarily produce symmetric design concepts when they are applied to asymmetric design embryos, as it is shown in Figure 57. In this case, rule 19, or in binary form 00010011, was applied to a symmetric design embryo (left) and an asymmetric design embryo (right). It is clear that even this symmetric design rule develops an asymmetric design concept when applied to an asymmetric embryo.

In the design experiments reported in this section, the entire set of 64 symmetric design rules was investigated. Each rule from the set was applied to 8 symmetric design embryos shown in Figure 58 and generated 8 design concepts of wind bracing systems from them. As before, the experiments were conducted using both groups of wind bracing elements. The developed design concepts were evaluated using both the first-order and the P-Δ analysis. The values of the total weight of the steel structural system and its maximum horizontal displacement calculated during these analyses were recorded.

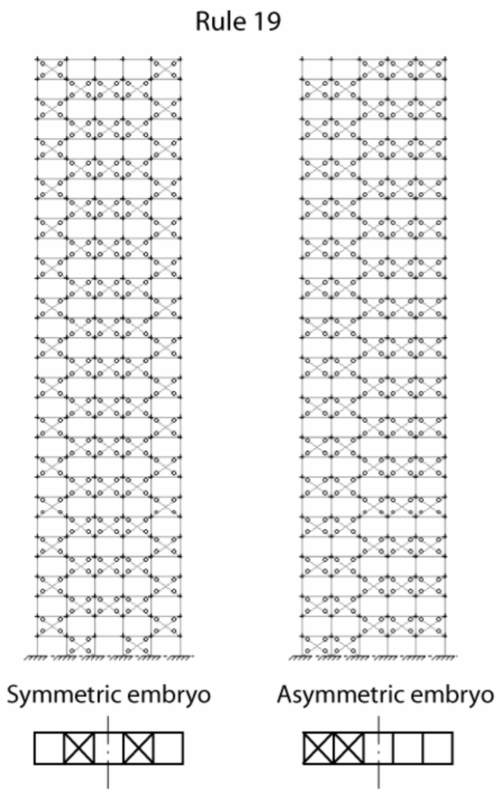


Figure 57. Design concepts developed from the symmetric rule 19 when a symmetric design embryo is used (left) and an asymmetric design embryo is used (right)

Best Designs

Table 35 shows the best results generated by elementary CA rules with the symmetry constraint. The three best design concepts developed by elementary CA rules with symmetry constraint were produced by rule 50. They exhibit various variations of the previously identified checkerboard pattern. The differences among the three design concepts occur only in the lowest part of the steel structure (3 lowest stories) due to different configurations of the design embryos used in the experiments. In fact, rule 50 developed the checkerboard pattern starting from all design embryos shown in Figure 58 except for two extreme cases when the design embryo consisted of all no bracings (empty cells) (see Figure 58a)) and all simple X bracings (see Figure 58h)). All design concepts developed by rule 50 from symmetric design embryos are shown in Table 36.

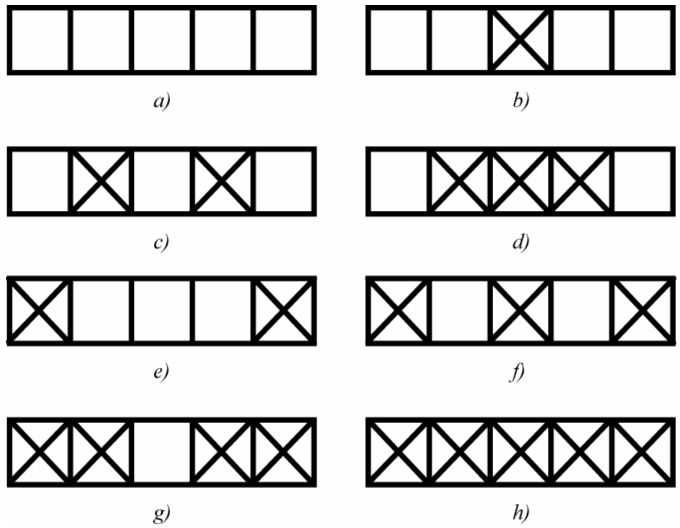


Figure 58. Eight types of symmetric design embryos used in the experiments with symmetric design rules

Symmetric Designs vs. Asymmetric Designs

When we compare the best symmetric design concepts (see Table 35) with the ones generated from random design embryos (see Table 31), we observe that there are no significant differences in their performance. The overall best design concept was produced from a randomly generated design embryo by rule 154 (see Table 31) but two best symmetric designs shown in Table 35 outperform the second best design produced from a random design embryo. Only 2 out of 12 design concepts shown in Table 31 are symmetric, namely the 7th and the 8th best designs developed by rules 19 and 178, respectively. In fact, the same design concepts can be found in Table 35. Rule 19 generates the 4th best symmetric design while rule 178 generates the 7th best symmetric design.

K bracings

The best design concepts produced by elementary CA rules with the second group of wind bracing elements are shown in Table 37. All of them exhibit the fully braced pattern in which all cells from the second story up to the topmost story of the structural system are occupied by K bracings. The differences among the design concepts shown in Table 37 are limited to the configurations of the design embryo.

The total weight of the best design concept was equal to 449,376 lbs. (calculated using the first-order structural analysis) and was slightly better than the weight of the best design concept produced in the experiments with randomly generated design embryo (see Table 33). In general, the best design concepts produced in the experiments with the symmetry constraint were of similar, if not identical, performance as the best design concept produced with randomly generated design embryos.

Table 35. Best symmetric designs of wind bracing systems consisting of simple X bracings produced by elementary CA rules with the symmetry constraint

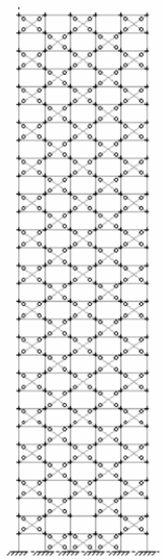
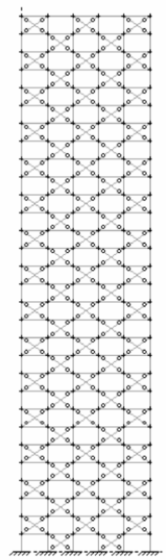
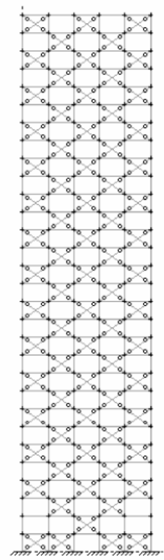
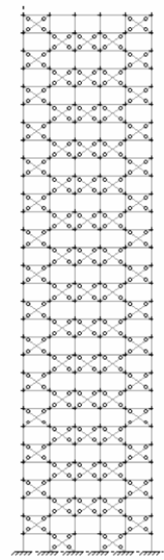
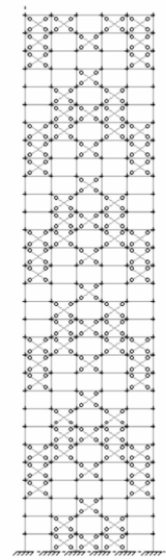
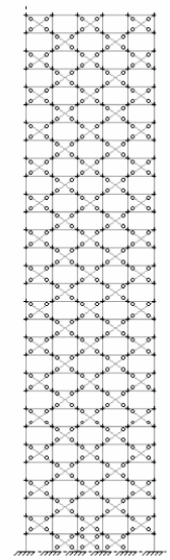
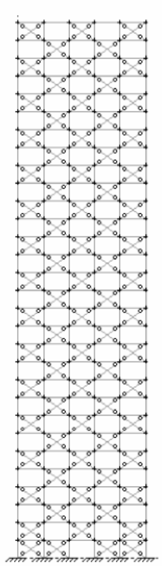
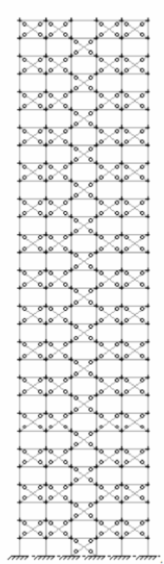
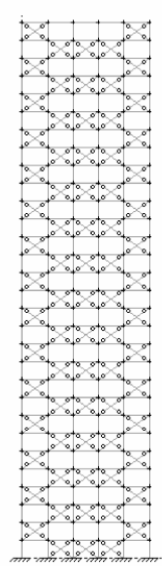
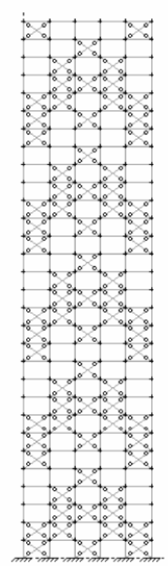
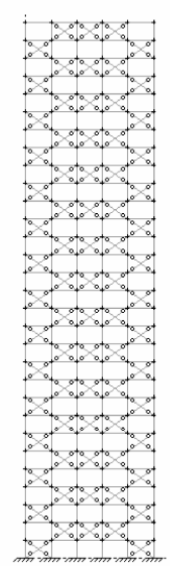
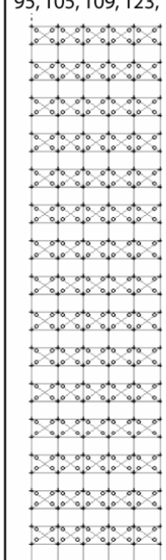
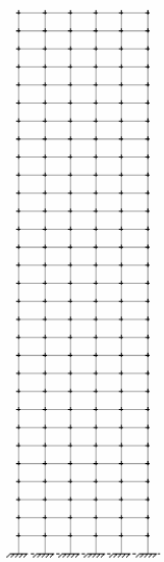
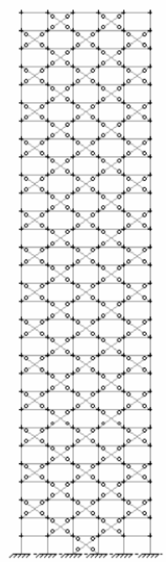
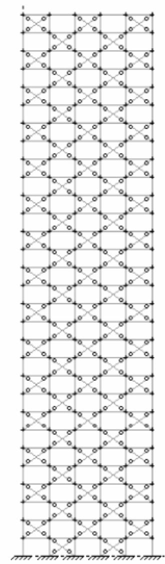
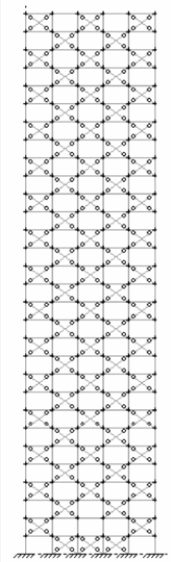
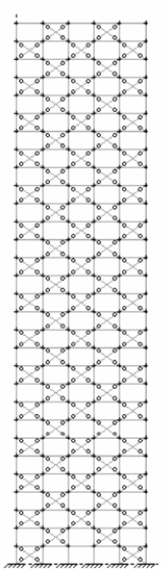
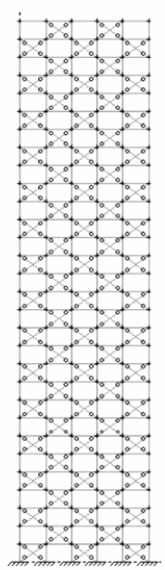
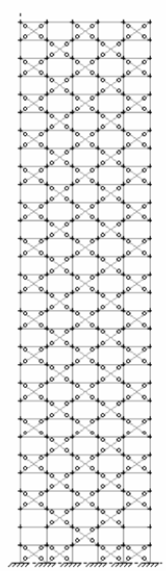
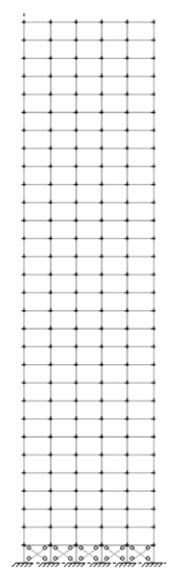
Rule 50	Rule 50, 51, 178, 179	Rule 50	Rule 19	Rule 105	Rule 178, 179
					
556,177 558,174 4.4727 4.5391	556,411 559,751 4.4602 4.5090	558,623 559,860 4.4957 4.5587	559,892 559,892 5.3018 5.3018	559,982 560,648 4.8465 4.9506	558,757 560,920 4.4522 4.5169
Rule 178, 179	Rule 51	Rule 19, 23, 51, 55	Rule 105	Rule 19, 23, 51, 55	Rule 1, 5, 19, 23, 33 37, 51, 55, 73, 77, 91, 95, 105, 109, 123, 129
					
559,935 562,465 4.446 4.5111	560,646 562,570 4.9940 5.1378	561,207 563,703 5.2940 5.4064	564,609 565,674 4.7397 4.8342	568,103 567,604 5.1049 5.2377	563,865 568,321 6.6963 6.8747

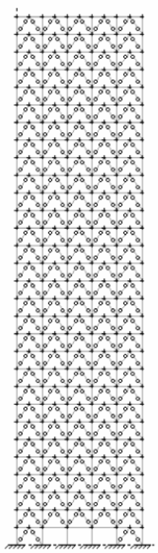
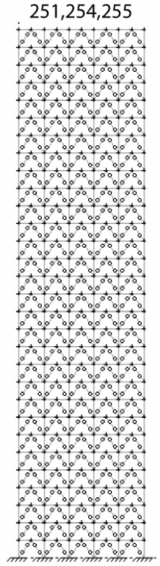
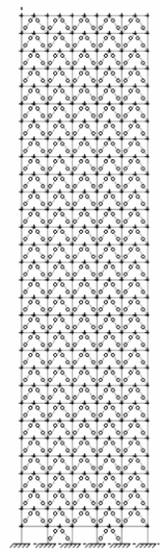
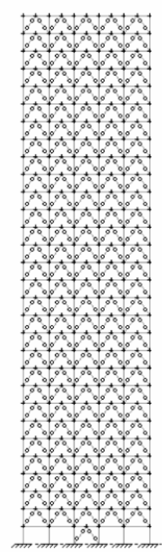
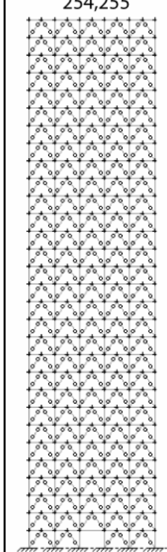
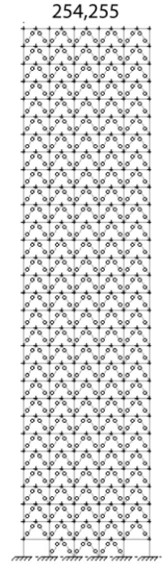
Table 36. Symmetric designs of wind bracing systems produced by elementary rule 50

Rule 50	Rule 50	Rule 50	Rule 50
			
4,684,228 4,684,228	568,951 570,251	556,411 559,751	556,177 558,174
1.0542 1.0594	4.3405 4.4106	4.4602 4.5090	4.4727 4.5391
Rule 50	Rule 50	Rule 50	Rule 50
			
573,728 571,427	576,124 573,823	558,629 559,860	4,891,396 4,891,396
4.3085 4.4072	4.2877 4.3860	4.4957 4.5587	1.0256 1.0306

Configuration of the Design Embryo

Both Table 35 and Table 37 show that better design concepts are developed from more general configurations of the design embryo than the simplest design embryo studied in section 6.2.1. In fact, only 1 out of 12 best design concepts presented in Table 35 and 1 out of 6 best design concepts shown in Table 37 were generated from the simplest design embryo consisting of either a single simple X bracing or K bracing located in the middle bay. The remaining design concepts presented in Table 35 and Table 37 were developed from more general configurations of the design embryo. Thus, **it is beneficial to employ more complex configurations of the design embryos in order to generate design concepts of better performance**. This result exactly corresponds to the findings discussed in the previous section in which I found that it is not sufficient to use the simplest design embryos and search only the space of the design rules. One should rather search both the space of design embryo configurations and the space of the design rules applied to these embryos.

Table 37. Best symmetric designs of wind bracing systems composed of K bracings and produced by elementary CA rules with the symmetry constraint

Rule 219,223,251,255	Rule 147, 179, 232, 233, 236, 237, 250, 251,254,255	Rule 182,183,254,255	Rule 151,183, 223,255	Rule 232,233, 236, 237, 250, 251, 254,255	Rule 218, 219, 222, 223, 250, 251, 254,255
					
449,376 452,620 4.9888 5.0727	458,274 452,748 4.8721 5.0615	449,521 452,766 4.9844 5.0680	450,234 453,021 4.9863 5.0781	450,234 453,021 4.9863 5.0781	450,234 453,021 4.9863 5.0781

6.2.4. Summary

In this section, I empirically investigated the simplest generative representations of wind bracing systems based on elementary cellular automata. These representations consist of a design rule based on an elementary CA rule and a design embryo determining the initial configuration of wind bracings.

In the first subsection, I described the results of design experiments in which I exhaustively searched the space of the design rules and applied them to an arbitrarily assumed design embryo. The simplest configuration of the design embryo was assumed which consisted of a single simple X bracing, or K bracing, located in the central bay. Even these extremely simple experimental settings were able to produce novel structural shaping patterns of good performance. I also compared the design concepts of wind bracing systems developed by elementary CAs with the design concepts generated randomly and found out that the former perform better in terms of both the total weight of the steel structural system and its maximum horizontal displacement. I compared the developed design concepts with the designs known from the structural engineering literature. I discovered that many traditionally known structural shaping patterns could be generated by elementary CA rules.

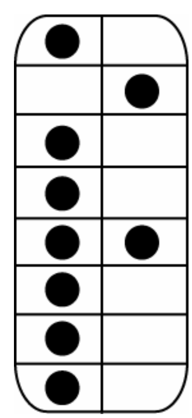
Furthermore, I investigated the impact of various representation specific parameters on the quality of generated design concepts. I found that the location of the design embryo (bottom vs. top of the steel structure) has on average no influence on the performance of the produced design concepts. On the other hand, the use of nonperiodic boundary conditions may increase the total weight of the steel structural systems by several percent when K bracings are used and has no impact on the quality of produced design concepts when X bracings are employed.

In the second subsection, I empirically studied more general configurations of the design embryo. Here, the design embryos were no longer restricted to assume the simplest possible configuration but were instead generated randomly. The results of the experiments have shown that these more complex configurations of the design embryo produced better results. This result shows that both the space of the design embryos and the space of the design rules should be searched concurrently.

In the third subsection, I demonstrated how we can incorporate domain knowledge in the generative representations. I illustrated that with the symmetry requirement frequently applied in structural design. I showed how we can constrain both components of the generative representation, i.e. the design embryo and the design rule, so that it develops symmetric design concepts. I also described the results of the design experiments with the symmetry constraint. They showed that on average no performance gain is achieved (in terms of the total weight of the structural system) when the symmetry constraint is imposed compared to the situation when no symmetry constraint is used and the design concepts are developed from random design embryos. In these experiments, however, only a single objective measure was employed to estimate the quality of the produced design concepts. It might be the case that the results will be different when more complex evaluation models will be assumed. The design experiments with the symmetry constraint also confirmed the previous results that more complex configurations of the design embryo produce on average better results.

In the following sections, I will further investigate more general representations of wind bracing systems. First, I will extend the number of types of wind bracing elements to more than 2. Then, I will experimentally investigate the design rules based on both standard and totalistic CAs (see explanations in section 2.2). Finally, I will empirically study the design rules based on two-dimensional CAs which are applied to 2D design embryos.

6.3. Design Concept Generators Based on 1D Cellular Automata

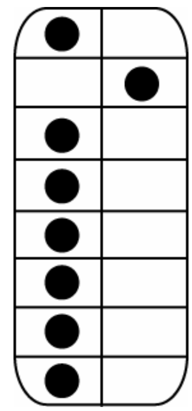


All experiments reported in section 6.2 considered only two types of wind bracing elements at a time. For these types of problems, elementary CAs were adequate to generate design concepts of wind bracing systems. However, for many design problems, we cannot restrict the design space to only two types of structural elements. On the contrary, majority of structural elements considered in 'real-world' design problems will have more than 2 possible types. From the representational point of view this corresponds to attributes having multiple values (see for example Figure 19 which graphically illustrates the values of attributes representing wind bracing elements with 7 possible values).

For these types of problems design concept generators based on elementary CAs are not sufficient and more general CAs must be used. In this section, one-dimensional cellular automata (1D CAs) are studied where each cell may have in general more than two values. One can also vary the size of the local neighborhood and thus control the radius of local interactions among cells.

Two types of 1D CAs have been studied. First, section 6.3.1 investigates standard 1D CAs as design concept generators of wind bracing systems in tall buildings. Here, cells representing wind bracing attributes have 7 possible values as shown in Figure 19. The following section explores the space of design rules based on totalistic 1D CAs (see section 6.3.2).

6.3.1. Standard 1D Cellular Automata



In this section, the results of the experiments are reported in which standard 1D CA were used to develop design concepts of wind bracing systems. Each cell had 7 possible values representing 7 types of wind bracing elements: no bracing, diagonal bracing \, diagonal bracing /, K bracing, V bracing, simple X bracing, and X bracing (see Figure 19).

As discussed earlier in section 2.2.1, increasing the number of possible cell values causes a rapid growth in the number of possible 1D CA rules. For example, when there are only 2 possible cell values then the size of the rule space is equal to $2^{2^3} = 256$ rules. When we increase the number of cell states (and keep the same size of the local neighborhood, i.e. equal to 3) to 7 then there are $7^{7^3} = 7^{343} = 7.4 \cdot 10^{289}$ possible 1D CA rules. When we also increase the radius of the local neighborhood to 2, then the rule spaces become even larger. In this case, there are $7^{7^5} = 7^{16807} = 3.6 \cdot 10^{14203}$ possible 1D CA rules!

When the radius of the local neighborhood is equal to 1 and the number of cell states is equal to 7, then there are 343 (i.e. $7 \cdot 7 \cdot 7$) possible combinations of cell values in the local neighborhood of size 3 compared to 8 possible combinations corresponding to binary cell values. Thus, assuming a fixed ordering of the local neighborhoods, we can represent any 1D CA rule with 7 possible cell values and the radius equal to 1 as a string of 343 digits. Each digit in this string can have a value from 0 to 6. The string contains the outcome values determined by a 1D CA rule and, given the assumed ordering of the local neighborhoods, uniquely defines each 1D CA rule. Similarly, we can represent any 1D CA rule with 7 possible cell values and the radius equal to 2 as a string of 14,203 digits.

In any case, the size of the 1D CA rule space with 7 possible cell values is truly enormous. It is impossible to search this space exhaustively, as I did in the previous sections with elementary CAs. Hence, only a random search of this vast space was conducted and its results are reported in this section.

Table 38 shows the parameters and their values used in the design experiments reported in this section. As stated earlier, 1D CAs with 7 cell values representing 7 types of wind bracings were used. A randomly selected design rule was applied to a randomly generated design embryo and developed a design concept of a wind bracing system from it. Two radii of the local neighborhood were studied: 1 and 2. They correspond to the sizes of the local neighborhood equal to 3 and 5, respectively. All experiments reported in this section used CA rules with periodic boundary conditions.

Two samples of 10,000 design concepts each (one sample for each radius length) were developed in this way and evaluated using the first-order analysis. The values of the total weight of the structural system and its maximum horizontal displacement were recorded. The results of the experiments are presented below separately for each radius.

Table 38. Parameters and their values used in the design experiments with 1D CAs

Experimental Parameter	*Value(s)*
Number of cell values	7
Radius of the local neighborhood	1, or 2
Boundary conditions	Periodic
Embryo generation mechanism	Random
Design rule search mechanism	Random
Random sample size	10,000

Best Designs

Table 39 shows 12 best design concepts developed by 1D CA rules when the radius of the local neighborhood was equal to 1. Four best designs presented in Table 39 exhibit various stages of development of a fully braced pattern in which either K bracings or V bracings were used. The best design concept exhibits the most developed fully-braced pattern covering the majority of the height of the structure.

As I discussed earlier, each design rule based on a 1D CA rule with 7 possible cell values can be expressed as a number consisting of 343 digits in base 7 or, when we use the numbering scheme introduced in section 2.2.2, as a number in base 10. Thus, using this convention we can represent the design rule that developed the best design concept shown in Table 39:

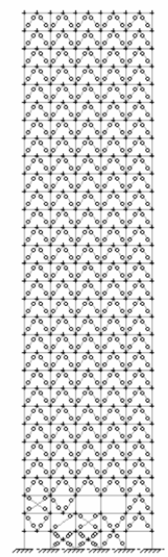
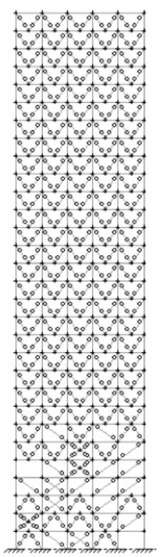
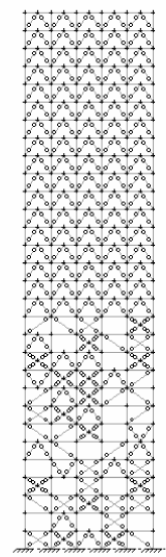
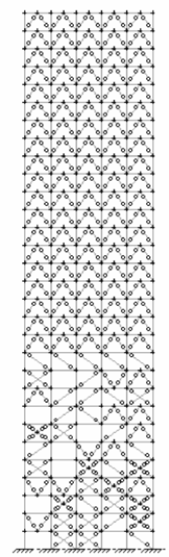
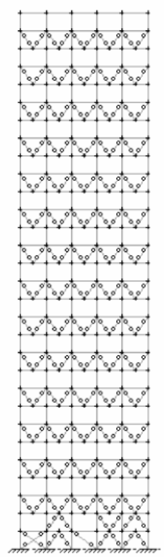
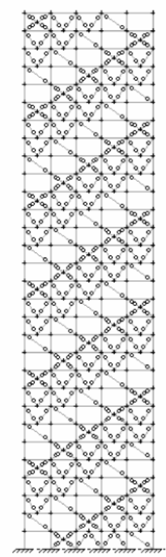
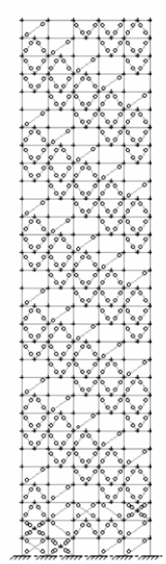
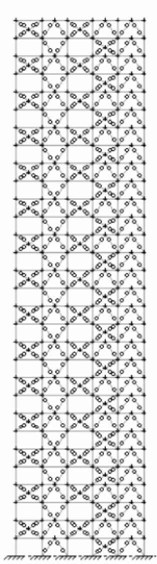
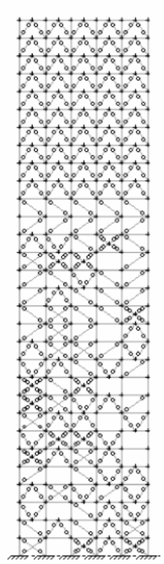
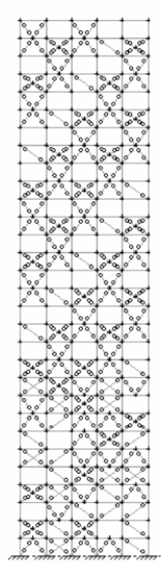
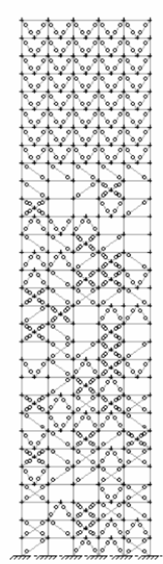
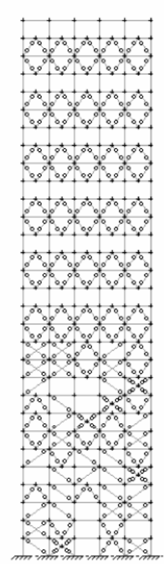
- In base 7 (343 digits):
  5454240166423423204241651314063660224464022342650214104236140421210513312216322540106643002214136324105261104655441333235305332413322336043552052334015135163641513135011163642501653534411234666404126661314012004132355535402104025603151540025054662640240542366312055300546356543156022355051311542240044113016304213122315110415253064214554553504
- In base 10:
  6002776407492514901114617073398615684374084324314311121424529643936523490727907489878656669572153303318637094296330646576910930169995083974251876572297312170522110265843651187227385489971283680283239391578757557821570667936196220575426885808438814277599624856639049900762088223216930307 8198

When we know the number of the design rule and the configuration of the design embryo, i.e. in this case the string consisting of 5 digits - 06640, we can uniquely define a design concept developed by this rule.

Structural Shaping Patterns

Table 39 contains several interesting structural shaping patterns of good performance. First, the 5th and the 12th best designs concepts exhibit two variations of the horizontal truss pattern. The former pattern was formed by V bracings located every two stories and it covers almost the entire height of the structural system. The latter one was generated by an interesting combination of K and V bracings which form two-story horizontal trusses. The two-story trusses cover more than half of the height of the structural system. Second, the 6th, 7th, and 10th design concepts exhibit elaborate versions of the macro bracing pattern. Here, the widths of the macro bracing patterns are equal to either 3 or 4 stories. The macro bracings are formed from a combination of various types of wind bracing elements, e.g. diagonal /, K, and V bracing in the case of the 7th design concept. Finally, the 8th design concept exhibits a new pattern which is formed by a combination of simple X bracings, V bracings, and K bracings.

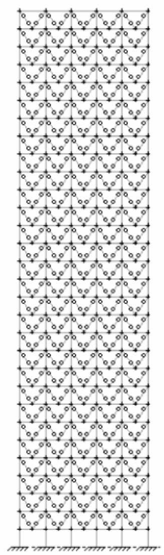
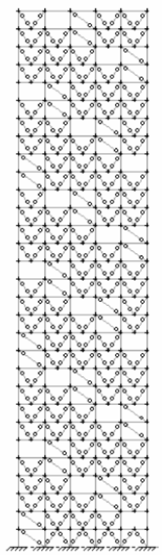
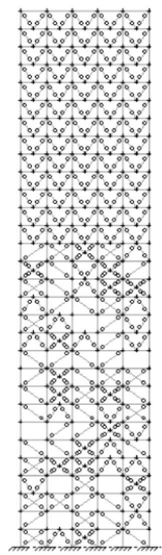
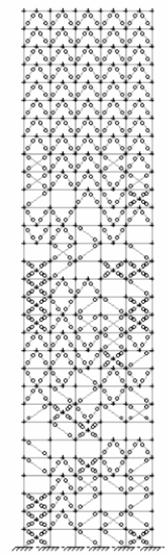
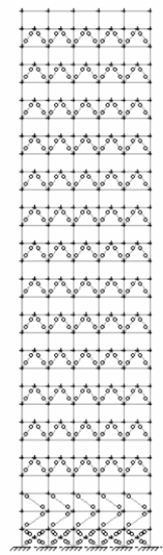
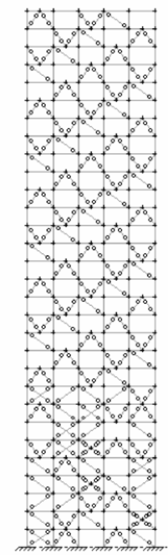
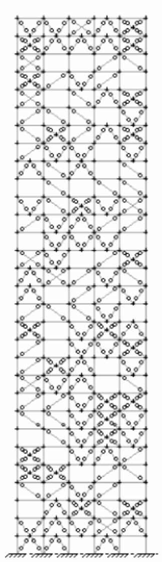
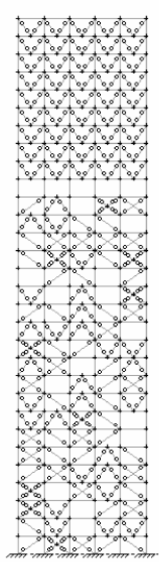
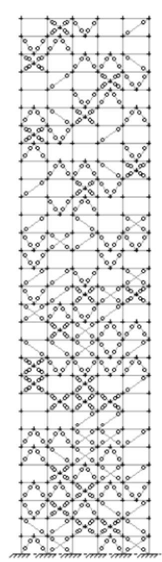
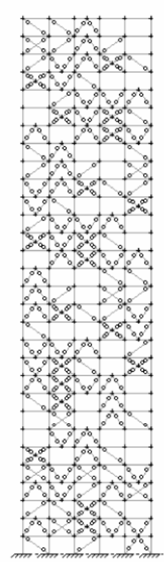
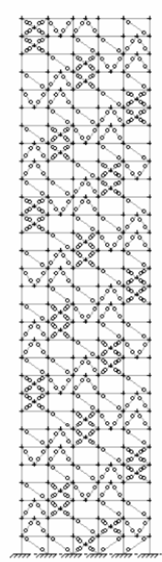
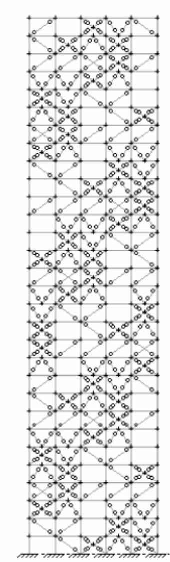
Table 39. Best designs of wind bracing systems developed by 1D CAs with the radius of the local neighborhood equal to 1

 465,359 4.8904	 481,499 5.3595	 497,998 4.7901	 498,787 4.7557	 509,992 7.9457	 518,027 5.7043
 519,774 5.5829	 520,363 5.8722	 522,723 4.8262	 525,560 5.6773	 525,975 4.9285	 526,312 6.4417

Impact of an Increased Size of the Local Neighborhood

In the second group of design experiments, the impact of the increased size of the local neighborhood was investigated. The best design concepts produced by 1D CA rules with the radius equal to 2 are shown in Table 40.

Table 40. Best designs of wind bracing systems developed by 1D CAs with the radius of the local neighborhood equal to 2

 454,304 5.7042	 522,330 5.8783	 522,703 4.7865	 524,800 5.0616	 527,780 7.1674	 528,501 5.4936
 528,907 5.2414	 528,980 5.1500	 530,263 5.3339	 530,282 5.2603	 530,298 5.6376	 531,067 5.1372

In this case, the best design concept exhibits the fully-braced pattern formed by V bracings which spans the entire height of the structural system. The total weight of the steel structure is comparable to the best design found in the experiments with K bracings (see Table 16). Various stages of the development of this pattern can also be found in the 3rd, 4th, and 7th design concept shown in Table 40.

The macro bracing patterns can be identified in the 2nd, 6th, 11th, and 12th design concept. The first two design concepts exhibit relatively simple versions of this patterns consisting of either V and diagonal \ bracings, or V, K, and diagonal \ bracings. The latter two macro bracing patterns are much more complex and formed by all 7 types of wind bracings elements.

Finally, the 7th, 9th, and 10th design concepts exhibit apparently chaotic patterns where no regularity can be found. It is worth mentioning that even the design concepts with the 'chaotic' patterns have better performance than the best design concepts composed of produced simple X bracings and by elementary CA rules.

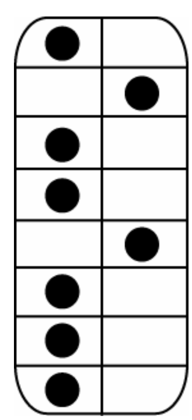
One-dimensional CAs vs. Elementary CAs

When we compare the performance of the design concepts generated by 1D CA rules (see Table 39 and Table 40) with the ones developed by elementary CA rules (see Table 8 and Table 16), we observe that **they are better than designs produced by X bracings but worse than designs consisting of K bracings**. This statement, however, cannot be generalized too far because the two rule spaces were not sampled equally. In the case of elementary CA rules, the entire rule space has been exhaustively searched. In the case of 1D CA rule, only a tiny portion of the design rule space was sampled.

I demonstrated earlier in this section that the rule space is enormous even when the smallest radius of the local neighborhood is used. It grows even more rapidly when we increase the radius. Thus, it is impossible to search this space exhaustively. We need to find ways to improve our possibilities of identifying good design rules in these vast rule spaces.

One of possible ways to achieve this goal is described in the following section. It discusses the use of totalistic 1D CAs instead of standard 1D CAs. Totalistic 1D CAs significantly reduce the size of the rule spaces. Another possibility is described in chapter 8, in which more intelligent search mechanisms (evolutionary algorithms) are used to search the vast rule spaces for good design rules.

6.3.2. Totalistic 1D Cellular Automata



In the previous section, I demonstrated that the number of 1D CA rules grows rapidly when we increase the number of possible cell values and/or the radius of the local neighborhood. There is a way, however, to substantially reduce the number of 1D CA rules by using totalistic 1D CA. The idea of a totalistic CA is to take the new value of each cell to depend only on the *average* value of the neighboring cells, and not on their individual values (see section 2.2.1 and Figure 6).

By using totalistic 1D CAs, we can reduce the size of rule space from $7^{7^3} = 7.4\cdot 10^{289}$ to $7^{3\cdot 7-2} = 7^{19} = 1.1\cdot 10^{16}$ when the radius is equal to 1 and from $7^{7^5} = 3.6\cdot 10^{14203}$ to $7^{5\cdot 7-4} = 7^{31} = 1.5\cdot 10^{26}$ when the radius is equal to 2. Thus, totalistic 1D CAs can reduce the size of the rule space by hundreds, or even thousands, orders of magnitude.

In the case of a totalistic 1D CA with 7 possible cell values, there are only 19, i.e. $3\cdot 7-2$, possible combinations of cell values in the local neighborhood of size 3. Similarly, there are

$5 \cdot 7 - 4 = 31$ possible combinations of cell values in the local neighborhood of size 5. Thus, by applying a totalistic 1D CA instead of a standard 1D CA, we reduce the representation of a design rule from 343 digits to only 19 digits and from 14,203 to 31 digits when the radius of the local neighborhood is equal to 1 and 2, respectively. Similarly as before, each of the 19 or 31 digits can have a value from 0-6. Given the assumed ordering of the local neighborhoods, each string of 19 or 31 digits uniquely defines a totalistic 1D CA rule.

Even though the space of totalistic 1D CA rules is significantly smaller than the space of standard 1D CA rules, it is still vast and cannot be searched exhaustively. Hence, as before, a random search of this rule space was performed. Randomly selected design rules based on totalistic 1D CAs were applied to randomly generated design embryos. Also in this case, two samples of 10,000 design concepts (one for each radius length) were produced in this way.

Best Designs

The best design concepts developed by totalistic 1D CA rules with the radius of the local neighborhood equal to 1 are shown in Table 41. All design concepts presented in the table exhibit the fully braced pattern in which either K bracings or V bracings were used. The only differences among the design concepts occur in the lowest part of the building (first 3 stories of the structural system).

When we compare the performance of the developed design concepts we observe that all of them outperform (in terms of the total weight of the steel structural system) the best design concept developed by standard 1D CA rules with the neighborhood radius equal to 1 (see Table 39). They are also better than all design concepts, except for the best one; generated by standard 1D CA rules with the radius equal to 2 (see Table 40). Thus, due to reduced size of the rule (search) space, totalistic 1D CA can much easier produce design concepts of good performance.

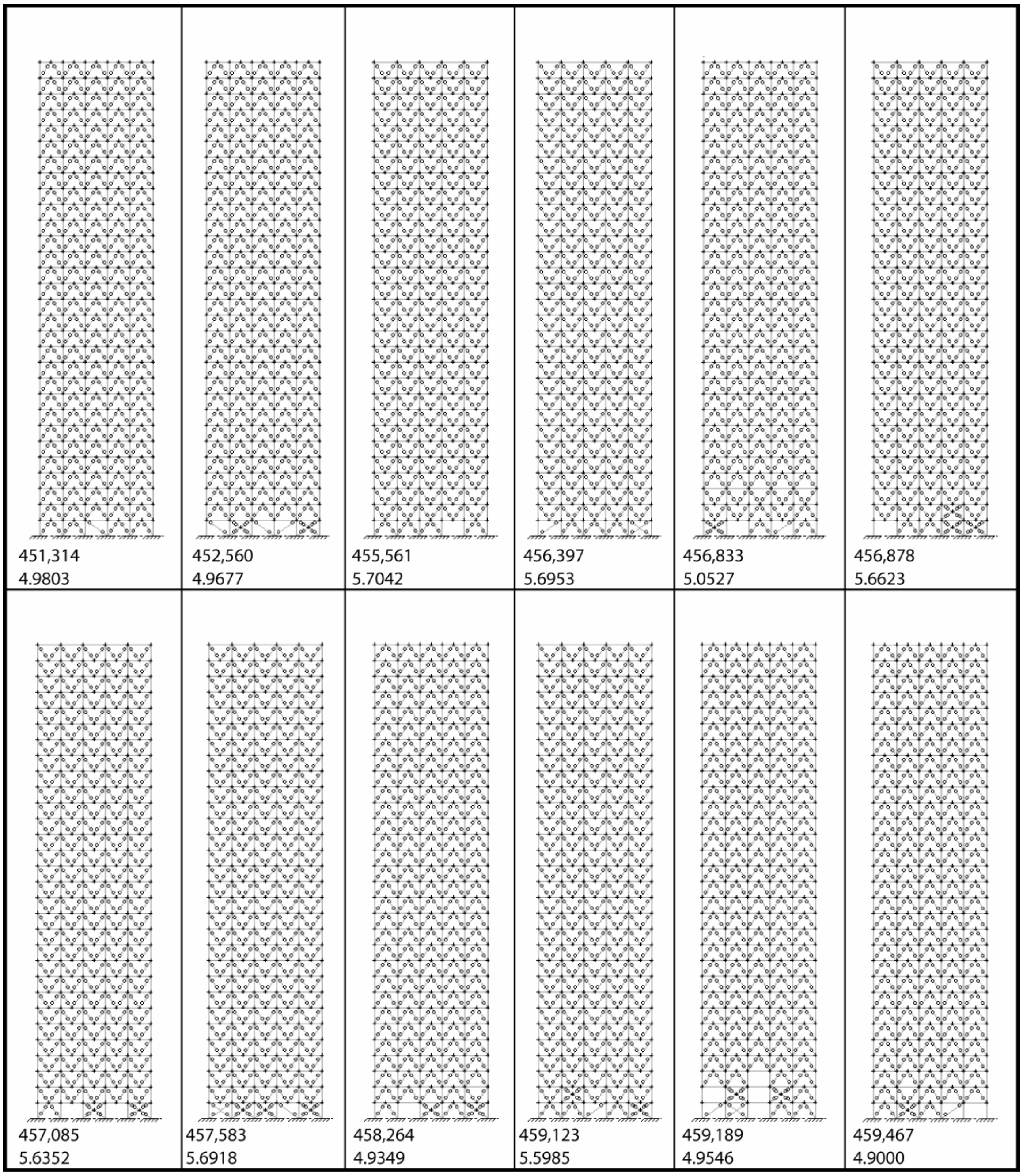
Structural Shaping Patterns

Totalistic 1D CA rules not only produced design concepts of good performance but also generated several interesting structural shaping patterns. The fully braced pattern outperformed other structural shaping patterns in terms of the total weight of the steel structural system. Moreover, it was produced by a relatively large number of design rules based on totalistic 1D CA rules. Hence, all design concepts shown in Table 41 exhibit this pattern. There were, however, many examples of the design rules that generated novel shaping patterns of good performance which were only slightly inferior to the designs exhibiting the fully braced pattern. Several such patterns are presented in Table 42.

Among the patterns shown in Table 42, we can find elaborate versions of the horizontal truss pattern in which multi-story trusses are formed. For example, the 9th and 10th design concepts presented in the table exhibit horizontal truss pattern in which trusses formed by a combination of X and K bracings span two stories of a structural system. The 2nd design in Table 42 contains three-story horizontal trusses produced by a combination of 3 types of wind bracing elements, namely K, V, and simple X bracings.

The 4th design shown in the table exhibits an interesting variation of the fully-braced pattern. Here, both V bracings and K bracings are used interchangeably every two stories. More complex patterns produced by a large number of wind bracing elements (virtually every cell is braced!) can be found in 4 design concepts shown in Table 42, i.e. in the 5th, 6th, 11th, and 12th designs. On the other hand, the 7th design contains very few wind bracing elements and nevertheless exhibits comparable performance.

Table 41. Best designs of wind bracing systems developed by totalistic 1D CAs with the radius of the local neighborhood equal to 1



Totalistic 1D CA rules with an increased size of the local neighborhood (the radius equal to 2) produced similar results to the ones reported above for the radius equal to 1. Hence, only the 3 best designs produced in these experiments are shown in Table 43 and a detailed discussion of the obtained results has been omitted.

Table 42. Interesting structural shaping patterns produced by totalistic 1D CA rules with the radius of the local neighborhood equal to 1

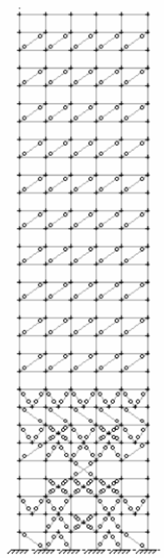
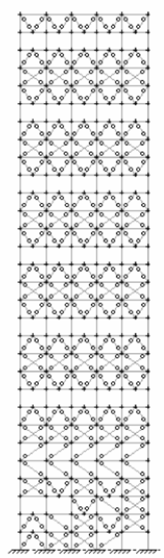
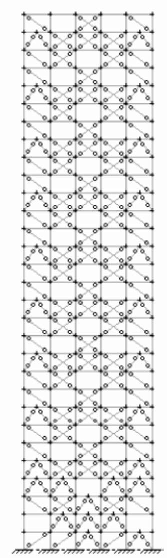
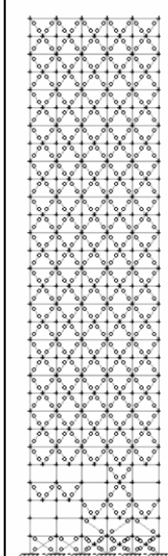
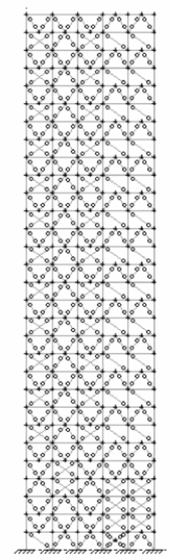
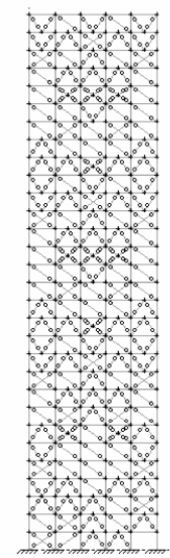
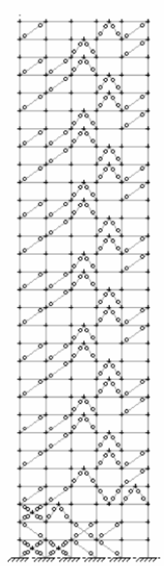
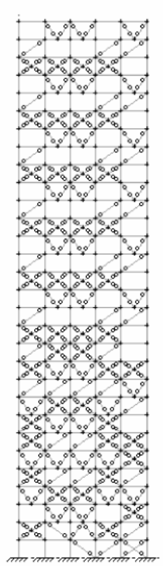
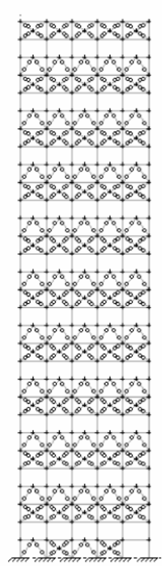
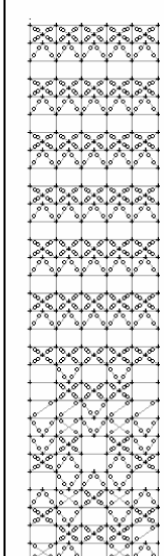
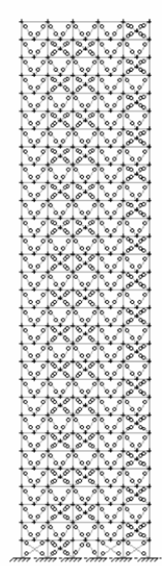
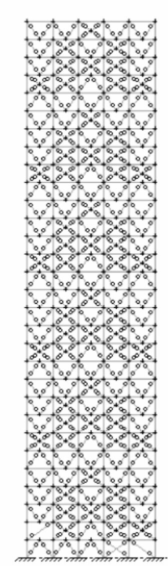
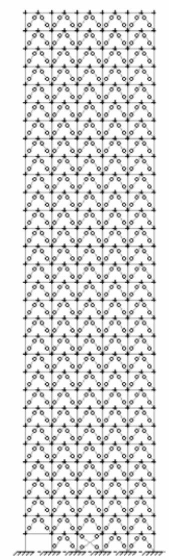
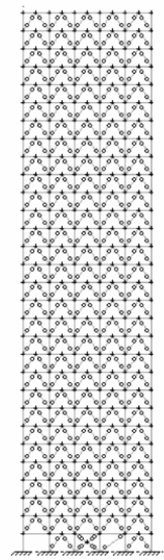
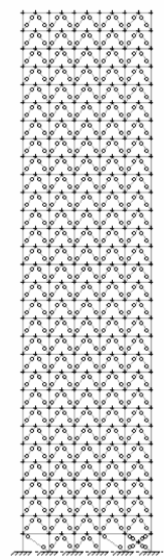
 691,578 6.4441	 556,913 6.1199	 570,597 4.7326	 523,576 8.3589	 567,959 5.1379	 562,779 5.0372
 530,063 6.2848	 516,741 5.6583	 515,081 7.5250	 516,794 6.5273	 517,991 5.2306	 519,587 4.9525

Table 43. Best designs of wind bracing systems developed by totalistic 1D CAs with the radius of the local neighborhood equal to 2

		
449,086 5.0213	451,018 4.9648	451,982 4.9759

6.3.3. Summary

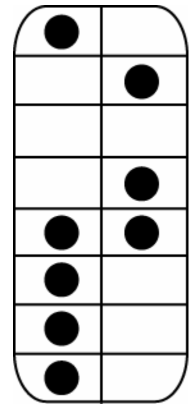
In this section, I extended the number of possible types of wind bracing elements to 7 and used more general types of cellular automata, namely 1D CAs to generate design concepts of wind bracing systems. Also, I tested the impact of an increased size of the local neighborhood (increased radius of local interactions) on the performance of the produced design concepts.

In the first subsection, I investigated the design rules based on standard 1D CA rules which work in a similar way as elementary CA rules studied in the previous section. I discussed a rapid growth of the sizes of 1D CA rule spaces when the number of cells states and the radius of the local neighborhood increase. This property prevents one from conducting an exhaustive search of the rule spaces, as it was the case with elementary CAs. Hence, only a random search in these vast spaces was performed.

The conducted experiments have shown that 1D CA rules with 7 types of wind bracing elements generate interesting structural shaping patterns. The patterns are even more elaborate than the ones discovered in the experiments with elementary CAs. The performance of the produced design concepts of wind bracing systems was better than the concepts composed of X bracings but worse than the ones consisting of K bracings. No significant impact of the increased radius of the local neighborhood on the performance of the design concepts was observed.

In the second subsection, I introduced the design rules based on totalistic 1D CA rules. Totalistic 1D CA rules substantially reduce the sizes of the rule spaces so that they can be searched more efficiently. The conducted experiments with totalistic 1D CA rules have shown that they not only generated intriguing structural shaping patterns but they also produced design concepts of significantly better performance than standard 1D CA rules. The best design concepts generated in these experiments are of comparable performance to the best design concepts produced by elementary CA rules and composed of K bracings (see Table 33) and much better than the concepts consisting of simple X bracings.

6.4. Design Concept Generators Based on 2D Cellular Automata



Design concept generators studied so far in this chapter investigated only one-dimensional cellular automata. The structural systems considered in this dissertation are, however, inherently two-dimensional. There also exist planar interactions among structural elements that are not accounted for using concept generators based on 1D CAs.

The underlying motivation behind the experiments reported in this section is that the planar interactions among structural members are important from the structural engineering/design point of view and should be explicitly modeled. In order to achieve it, two-dimensional cellular automata (2D CAs) have been proposed as design concept generators in section 4.4.3.

Similarly as it was the case with 1D CAs, there are two types of 2D CAs: standard and totalistic. Both types of 2D CAs have been investigated in this dissertation and the obtained results are reported in sections 6.4.1 and 6.4.2, respectively.

When using generative representations based on 2D CAs, one has to specify not only the radius of the local neighborhood (2D neighborhood in this case) but also its shape. Two most frequently used shapes of the local neighborhood in scientific literature include Moore neighborhood and von Neumann neighborhood (see section 2.2.1 and Figure 8). In this dissertation, a larger selection of shapes of the local neighborhood was studied, including Moore neighborhood, von Neumann neighborhood, diagonal neighborhood, north-south neighborhood, and east-west neighborhood (see Figure 59).

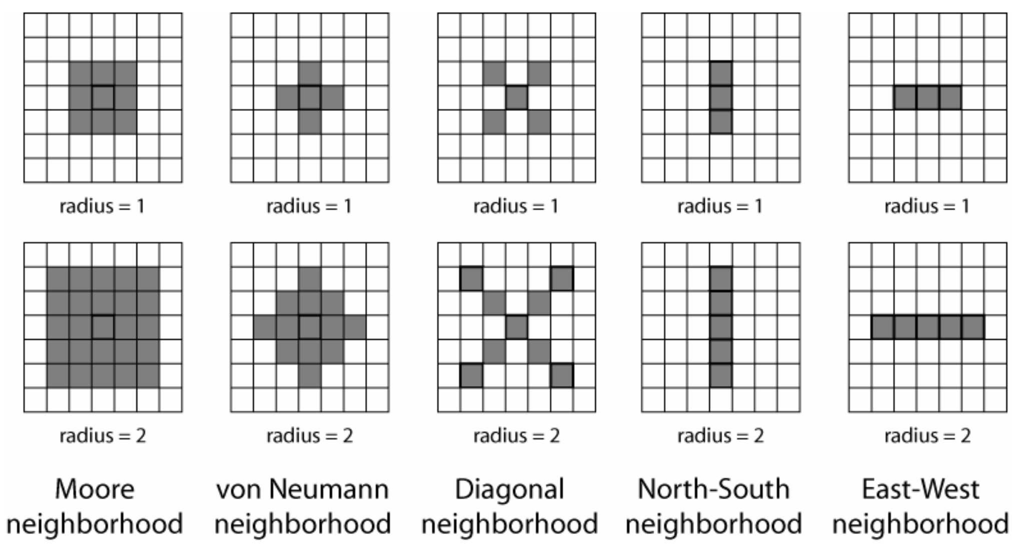


Figure 59. Shapes of the two-dimensional local neighborhoods used in the experiments with design concept generators based on 2D CAs

An additional experimental parameter that had to be specified for the generative representations based on 2D CAs is the number of iterations of a 2D CA rule. This parameter was introduced earlier in section 4.4.3 and named *iteration_max*. Several sensitivity analyses were performed to determine the impact of the value of this parameter on the performance of obtained results.

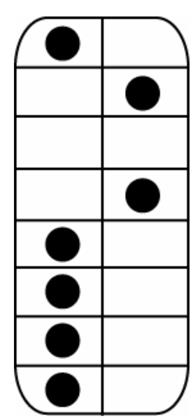
Table 44 shows the experimental parameters and their values used in the design experiments with 2D CAs. The number of types of wind bracing elements was restricted to 3 only: no bracing, K bracing, and simple X bracing in order not to cause an explosion of the size of the rule spaces (which were anyway vast). This resulted in a significant reduction of the search space. On one hand, a large number of potentially fit design concepts were omitted because they could not be represented. On the other hand, this assumption helped me focus on various aspects of representing spatial interactions among structural elements. They were investigated experimentally by varying the shape and radius of the local neighborhood of 2D CA rules.

Similarly as in the previous sections, a random sample of 10,000 design concepts was generated for each shape of the local neighborhood. Randomly selected design rules based on 2D CA were applied to randomly generated design embryos (now in the form of 2D configurations) of wind bracing elements (see section 4.4.3). The design rules were subsequently iterated the number of times equal to the value of the *iteration_max* parameter. At each step, a new design concept was produced and evaluated using the first-order structural analysis.

Table 44. Parameters and their values used in the design experiments with 2D CAs

Experimental Parameter	*Value(s)*
Number of cell values	3
Shape of the local neighborhood	Moore, von Neumann, diagonal, north-south, east-west
Radius of the local neighborhood	1 or 2
Embryo generation mechanism	Random
Design rule search mechanism	Random
Number of iterations of a 2D CA rule (*iteration_max*)	5, 10, 20, 50, 100, 1000
Random sample size	10,000

6.4.1. Standard 2D Cellular Automata



In this section, experimental results obtained using standard 2D CAs are presented. In these experiments, design rules based on standard 2D CA rules were applied to 2D design embryos and developed design concepts of wind bracing systems from them. As stated earlier, the number of types of wind bracing elements was restricted to 3 only. 5 shapes of the local neighborhood were investigated experimentally. Due to enormous sizes of the standard 2D CA rule spaces, only one length of the radius of the local neighborhood was investigated, i.e. the radius equal to 1. Obtained results are reported separately for each of the considered neighborhoods in the following subsections.

Moore Neighborhood

The experiments reported in this subsection involved randomly generated design embryos in a form of 2D configurations of wind bracing elements and randomly selected design rules based on 2D CA rules with Moore neighborhood (see Figure 59). The size of this 2D CA rule space was equal to $3^{3^9} = 3^{19683} = 1.5 \cdot 10^{9391}$ (cells had 3 possible values and the radius was equal to 1).

Assuming a fixed ordering of the local neighborhoods, we can represent any 2D CA rule with 3 possible cell values and Moore neighborhood as a string of 19,683 digits where each digit can have a value from 0 to 2. As described before, the string contains the outcome values determined by a 2D CA rule and, given the assumed ordering of the local neighborhoods, uniquely defines each such rule.

Several values of the *iteration_max* parameter, i.e. the parameter defining the number of iterations of a 2D CA rule, were investigated. Figure 60 shows typical results obtained in the design experiments with the value of *iteration_max* parameter equal to 5. In this case, the initial 2D configuration of wind bracing elements was generated randomly (*t=0*) and a 2D CA rule was applied 4 times. In this way, 5 design concepts were generated and subsequently evaluated.

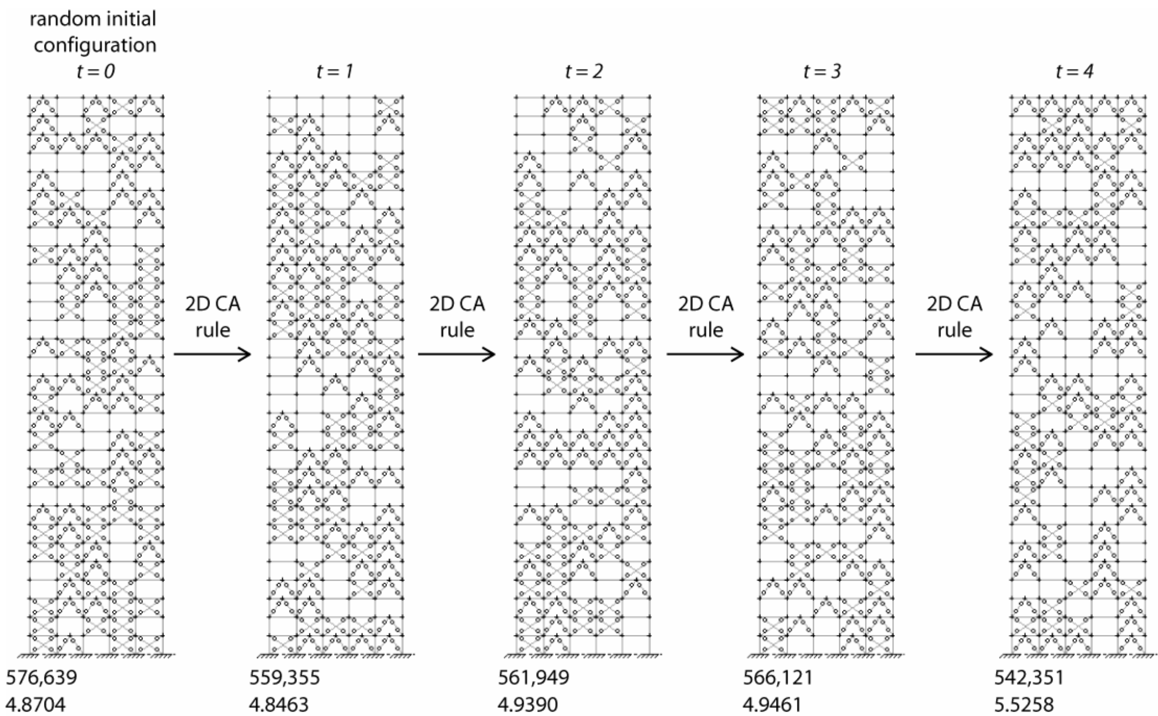
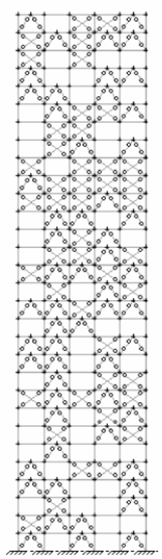
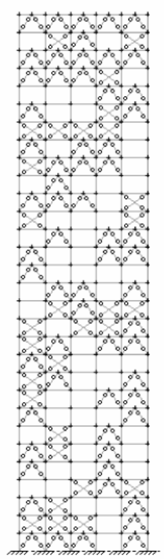
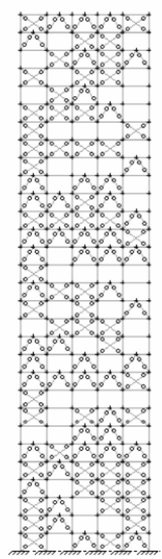
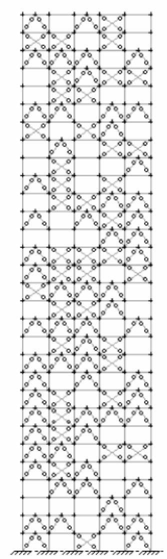
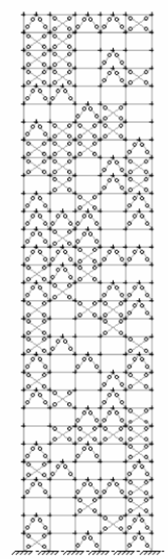
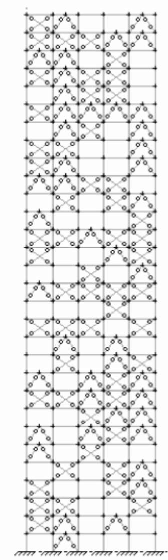


Figure 60. Design concepts of wind bracing systems generated by 5 iterations of a 2D CA rule with Moore neighborhood

Best Designs

The best design concepts developed by 2D CA rules with Moore neighborhood are shown in Table 45. The best design concept had the total weight equal to 537,086 lbs. whereas the median total weight of the entire sample of design concepts was equal to 570,751 lbs. It is difficult to identify any structural shaping patterns shared by several design concepts shown in Table 45. On the contrary, they rather exhibit randomly looking configurations of K and simple X bracings.

Table 45. Best design concepts of wind bracing systems developed by 2D CA rules with Moore neighborhood

					
537,086 5.3078	542,351 5.5258	544,452 5.0626	544,592 5.1822	544,895 5.0898	545,212 5.2109

Impact of the Number of Iterations

Also, no significant differences were detected in terms of the impact of the number of iterations (*iteration_max*) on the performance of generated design concepts. For example, the following median values of the total weight of the steel structural systems produced after the number of iterations specified by the value of the *iteration_max* parameter were obtained:

- 5 iterations: 570,229 lbs.
- 10 iterations: 574,777 lbs.
- 20 iterations: 571,047 lbs.
- 50 iterations: 568,925 lbs.
- 100 iterations: 574,188 lbs.
- 1000 iterations: 569,721 lbs.

In this case, the best median value was achieved for the design concepts generated after 50 iterations of a 2D CA rule with Moore neighborhood. However, the differences between the median values are small and insignificant. Hence, it is difficult to uniquely point out to a specific value of the *iteration_max* parameter which produces the best results for 2D CAs with Moore neighborhood.

Von Neumann Neighborhood

In this subsection, the same experimental parameters were used with one exception: von Neumann neighborhood (see Figure 59) was employed instead of Moore neighborhood. In this case, the size of the 2D CA rule space was significantly smaller and equal to $3^{3^5} = 3^{243} = 8.7 \cdot 10^{115}$ (but still very large!). Each 2D CA rule with 3 cell values and von Neumann neighborhood could be represented as a string of 243 digits (in 0-2 range), given the same assumption on the ordering of the local neighborhoods.

Best Designs

Table 46 shows the best designs produced by 2D CA rules with von Neumann neighborhood. The best design concept found had the total weight equal to 536,390 lbs. and was only slightly better (by 696 lbs. or 0.1 percent) than the best design concept produced using Moore neighborhood. The median total weight of the entire sample of design concepts was equal to 572,063 lbs., which is slightly more than the median value obtained for Moore neighborhood.

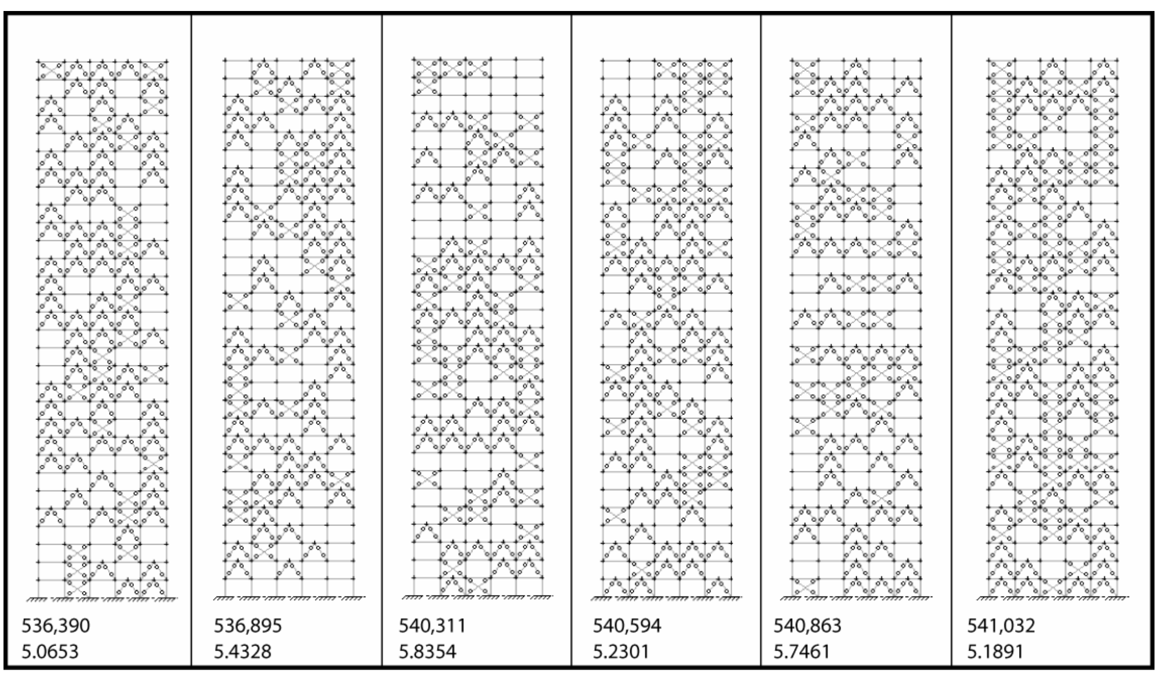
Structural Shaping Patterns

Similarly as it was the case with Moore neighborhood, most of the design concepts shown in Table 46 have randomly looking configurations of K and simple X bracings. There are, however, two exceptions. Namely, the 4th and 6th design concepts exhibit some forms of emergent macro bracing patterns, particularly in the lower and middle parts of the structural system. The 6th design concept has more wind bracing elements located in the outer bays of the structural system and hence, the macro bracing pattern is not readily visible.

Impact of the Number of Iterations

Half of the design concepts presented in Table 46 were generated in the experiments with 50 iterations of a 2D CA rule, 2 design concepts in the experiments with 10 iterations of the rule, and 1 in the experiments with 100 iterations of the design rule. The differences between the median values of the total weight of steel structural systems obtained after various numbers of iterations of 2D CA rules were, again, small. The lowest median (565,826 lbs.) was obtained when the *iteration_max* parameter was equal to 100 and the largest value (575,093 lbs.) was achieved when it was equal to 50.

Table 46. Best design concepts of wind bracing systems developed by 2D CA rules with von Neumann neighborhood



Diagonal Neighborhood

In this group of experiments, design rules based on 2D CA rules with the diagonal neighborhood (see Figure 59) were investigated. The diagonal neighborhood was selected because it explicitly models the interactions among the current cell and its neighbors located along the diagonals, a pattern that frequently occurred in so-called macro bracings.

In this case, the size of the 2D CA rule space was exactly the same as in the case of von Neumann neighborhood (but this is only true when the radius of the local neighborhood is equal to 1!), i.e. $3^{3^5} = 3^{243} = 8.7 \cdot 10^{115}$. Hence, each 2D CA rule with 3 cell values and the diagonal neighborhood could also be represented using a string of 243 digits.

Best Designs

Best design concepts obtained in this group of experiments are presented in Table 47. The total weight of the best design was equal to 542,354 lbs. and was slightly worse (by about 5,000-6,000 lbs.) than the best design concepts generated by 2D CA rules with Moore and von Neumann neighborhoods. The overall median total weight (572,311 lbs.) of the entire sample of design concepts was almost identical to the median values obtained for the two previously investigated shapes of the local neighborhood.

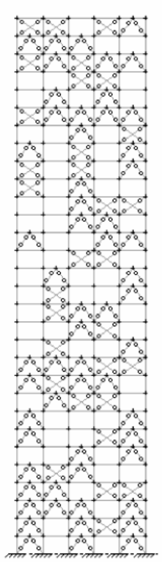
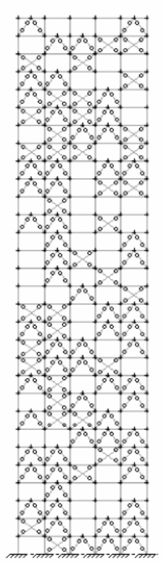
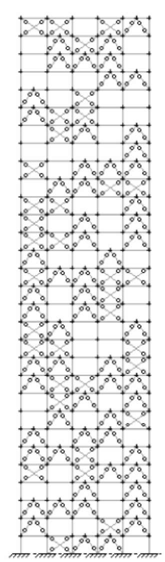
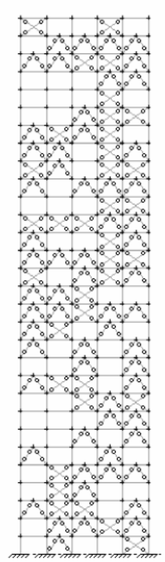
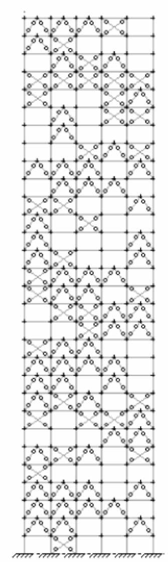
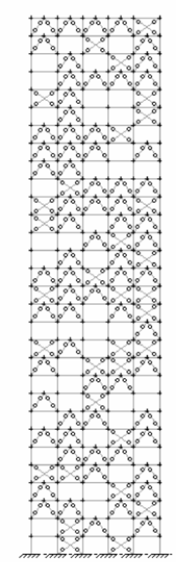
Structural Shaping Patterns

Table 47 also shows that there are no qualitative differences with respect to the structural shaping patterns exhibited by the best design concepts. As before, most of them show fairly random looking configurations of K and simple X bracing. In three cases, namely 2nd, 5th and 6th design concepts shown in Table 47, emergent macro bracing patterns are being formed mainly in the central part of the structural system.

Impact of the Number of Iterations

The differences between the medians obtained for various values of the *iteration_max* parameter were, as before, insignificant and oscillated between 568,485 lbs. (20 iterations) and 578,041 lbs. (1,000 iterations).

Table 47. Best design concepts of wind bracing systems developed by 2D CA rules with the diagonal neighborhood

					
542,354 5.4425	543,859 4.9895	544,183 5.0893	544,354 5.5634	544,846 5.3230	544,920 5.1654

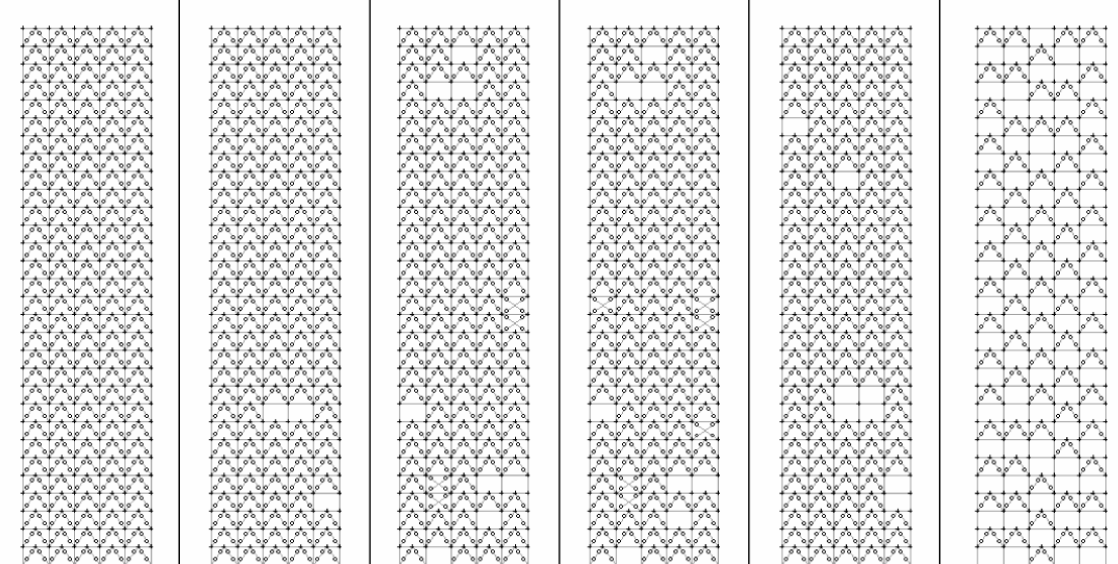
North-South Neighborhood

The design rules based on 2D CA rules with the north-south neighborhood explicitly modeled vertical interactions among wind bracing elements in a structural system. Here, the size of the rule space was equal to $3^{3^3} = 3^{27} = 7.6 \cdot 10^{12}$ (as before, 3 possible cell values were used). In this case, any 2D CA rule with the north-south neighborhood could be represented by a string of 27 digits.

Best Designs

Table 48 shows 6 best design concepts developed by 2D CA rules with the north-south neighborhood. The best design concept had the total weight equal to 458,274 lbs. and was significantly better than the best design concepts produced by 2D CA rules with other shapes of the local neighborhood described previously. In fact, all design concepts shown in Table 48 have better performance then any of the best design concepts generated using 2D CA rules with Moore, von Neumann, or diagonal neighborhoods.

Table 48. Best design concepts developed by 2D CA rules with the north-south neighborhood



458,274 4.8721	471,340 4.8376	491,032 4.8602	492,552 4.8508	496,053 4.8079	497,803 5.9123

The overall median value of the total weight of the entire sample of design concepts generated using 2D CA rules with the north-south neighborhood was equal to 573,875 lbs. and almost identical to the median values reported previously for various shapes of the local neighborhood. Also, as before, no significant differences were detected in terms of the performance of the design concepts generated using the range of values of the *iteration_max* parameter. The medians changed from 564,626 lbs. (100 iterations) to 575,601 lbs. (10 iterations).

Structural Shaping Patterns

Design concepts shown in Table 48 exhibit qualitatively different structural shaping patterns than the ones produced previously by 2D CA rules with other shapes of the local neighborhood. The 5 best design concepts have the fully-braced pattern composed of K bracings. The best design concept exhibits a fully developed pattern in which all structural cells are occupied by K bracings. The other 4 design concepts exhibit slight variations of the fully braced pattern in which some cells contain either simple X bracing or no bracings. The 6th best design concept shown in Table 48 has an interesting structural shaping pattern which is similar to (but not exactly the same) the checkerboard pattern identified previously in the design experiments with elementary CA. The exact checkerboard pattern has been developed only in the middle part of the structural system while two different variations of this pattern are visible in the lower and upper parts of the structure.

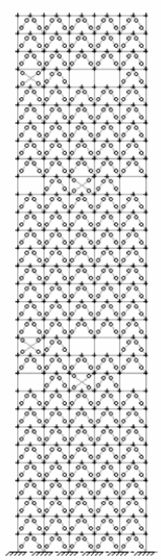
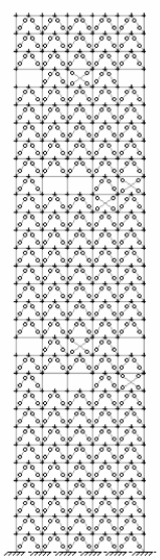
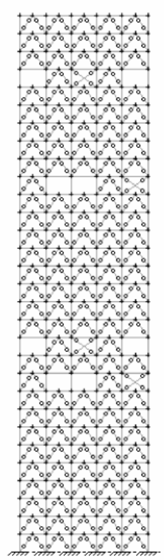
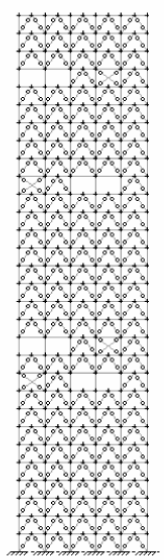
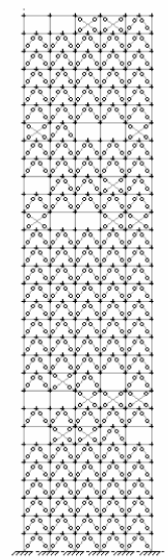
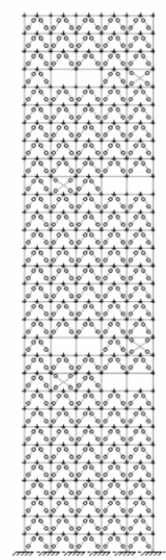
East-West Neighborhood

The east-west local neighborhood explicitly modeled horizontal interaction among the structural elements. In this case, the size of the rule space and the length of its representation, as a string of digits, were exactly the same as for the north-south neighborhood and equal to $3^{3^3} = 3^{27} = 7.6 \cdot 10^{12}$ and 27, respectively.

Best Designs

Best design concepts developed by 2D CA rules with the east-west neighborhood are shown in Table 49. The total weight of the best design concept found was equal to 476,944 lbs. and was worse than the total weight of the best design concept generated by the north-south neighborhood by more than 18,500 lbs. On the other hand, it was significantly better (by about 60,000 lbs.) than the total weight of the design concepts generated by Moore, von Neumann, and diagonal neighborhoods.

Table 49. Best design concepts developed by 2D CA rules with the east-west neighborhood

					
476,944	478,981	486,412	489,000	493,107	496,234
4.9092	5.0072	4.8949	4.9638	4.8980	4.7995

The overall median value of the total weight of the entire sample of design concepts generated using 2D CA rules with the east-west neighborhood was equal to 580,460 lbs. It was the highest (worst) value obtained in all design experiments with standard 2D CA rules. Also, the observed variations in median total weights of the design concepts developed for different values of the *iteration_max* parameter were the largest (more than 20,000 lbs.) and ranged from 567,947 lbs. (50 iterations) to 588,841 lbs. (100 iterations).

Structural Shaping Patterns

The structural shaping patterns shown in Table 49 are similar to the ones produced by 2D CA rules with the north-south neighborhood. All 6 design concepts presented in the table exhibit slight variations of the fully braced pattern in which most of the cells are occupied by K bracings. It is, however, interesting to observe the influence of the shape of the local neighborhood in this case. All disruptions of the uniform pattern produced by K bracings occur along the horizontal direction (within a single story of the tall building).

Summary

In this section, I empirically investigated generative representations based on standard 2D CAs. These representations consist of a design rule based on a standard 2D CA rule and a design embryo in a form of a 2D configuration of wind bracing elements.

Five subsections reported the results of the experiments in which I studied the influence of 5 different shapes of the local neighborhood on the performance of design concepts of wind bracing systems in tall buildings. I also investigated the impact of the number of iterations of a 2D CA rule, denoted in this dissertation by the *iteration_max* parameter, on the quality of produced design concepts. The experimental results reported in this section focused on the qualitative (patterns) and quantitative (best and median performance) aspects of the design generation processes.

In the first 3 subsections, I investigated Moore, von Neumann, and diagonal neighborhoods. The conducted design experiments have shown that these 3 neighborhood shapes produce generally inferior results in terms of the total weight of the design concepts compared to generative representations based on 1D CA. The best design concepts generated in these experiments exhibited mainly random looking configuration of K and simple X bracing. In several cases, however, some emergent macro bracing patterns were formed.

The situation was different for the remaining two shapes of the local neighborhood, namely the north-south neighborhood and the east-west neighborhood. The best design concepts generated by standard 2D CA rules with these two shapes showed good performance, which is comparable to the performance of the best design concepts generated by 1D CA rules.

These findings are illustrated graphically in Figure 61 which shows the median and best performance of the design concepts generated by 2D CA rules with 5 different shapes of the local neighborhood.

Figure 61 clearly shows that the differences among median values of the total weight of design concepts generated using various shapes of the local neighborhood are small and negligible. The differences, however, do occur for the best design concepts developed by 2D CA rules with the north-south and east-west neighborhoods which substantially outperformed the other 3 shapes of the local neighborhood.

Figure 62 illustrates the influence of the value of the *iteration_max* parameter on the median performance of the generated design concepts. It shows that there is no preferred value of this parameter which produces the best design concepts. On the contrary, for some shapes of the local neighborhood (von Neumann, north-south) a large number of iterations of a 2D CA rule was preferred while for other shapes (diagonal, east-west) smaller values produced better results.

In this figure, only the median values of the total weight of the structural systems were considered. The values of the *iteration_max* parameter had a significant impact of the dynamics of the design processes. Their influence on the dynamics of design processes also changed with the shape of the local neighborhood.

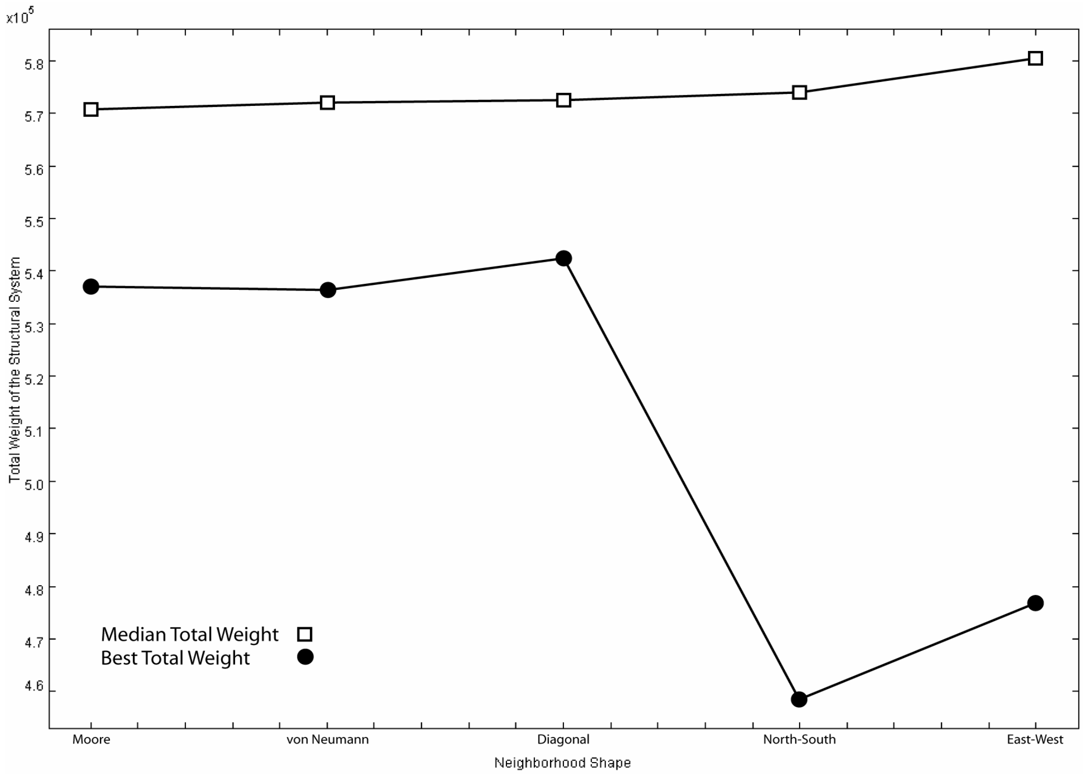


Figure 61. Comparison of the median and best performance of the design concepts generated by standard 2D CA rules with 5 shapes of the local neighborhood

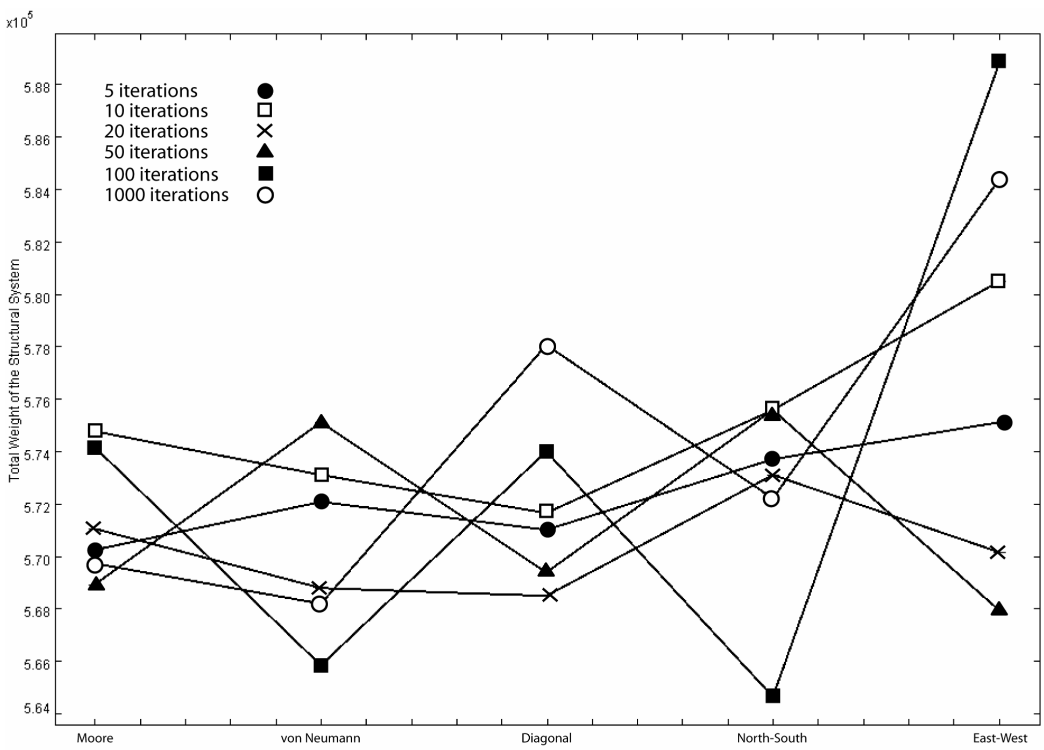
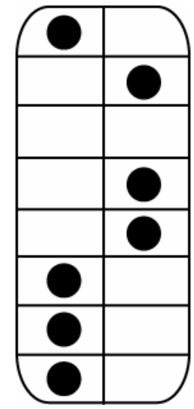


Figure 62. Impact of the value of the *iteration_max* parameter on the median total weight of generated design concepts for different shapes of the local neighborhood

6.4.2. Totalistic 2D Cellular Automata



As I showed in the previous section, the sizes of the standard 2D CA rule spaces are enormous even for moderate values of cell states (3) and radii of the local neighborhood (1). When a larger number of cell states needs to be considered or when the radius of the local neighborhood is larger than 1 then the only computationally feasible approach involves totalistic 2D CAs. In this section, I report the experimental results obtained using design rules based on totalistic 2D CA rules. Similarly as in the previous section, the design rules were applied to 2D design embryos and developed design concepts of wind bracing systems from them. Also, only 3 types of wind bracing elements were considered. This time, however, 2 values of the radius of the local neighborhood were studied experimentally, i.e. the radius equal to 1 and 2. As before, the obtained results are divided with respect to the shapes of the local neighborhood that were used in the experiments and reported in the following subsections.

Moore Neighborhood

The size of the totalistic 2D CA rule space with 3 possible cell values and Moore neighborhood was equal to $3^{9\cdot 3-8} = 3^{19} = 1.1\cdot 10^{9}$ when the radius of the local neighborhood was equal to 1 and $3^{25\cdot 3-24} = 3^{51} = 2.1\cdot 10^{24}$ when the radius equaled 2.

Thus, we can represent any totalistic 2D CA rule with 3 possible cell values and Moore neighborhood as a string of 19 digits and 51 digits, when the radius is equal to 1 and 2, respectively. This corresponds to the reduction of the length of the representation of a design rule by several orders of magnitude. In the case of standard 2D CA rules with the radius of 1 and 3 possible cell values we needed 19,683 digits whereas for a totalistic 2D CA rule with the same parameters we need only 19 digits.

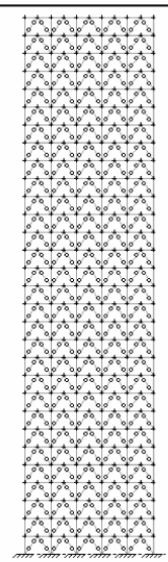
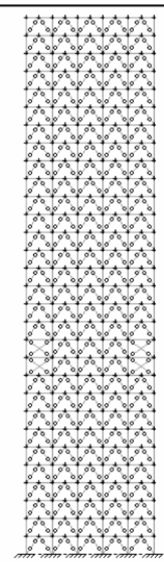
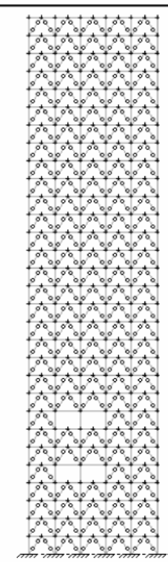
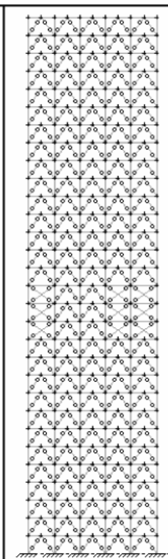
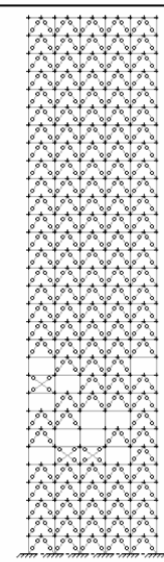
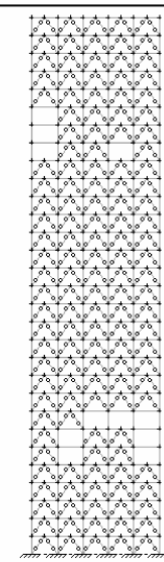
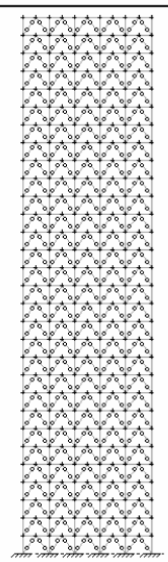
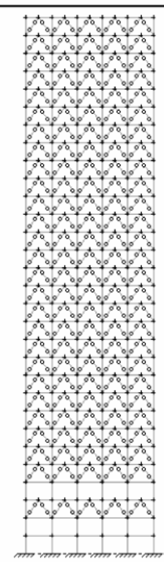
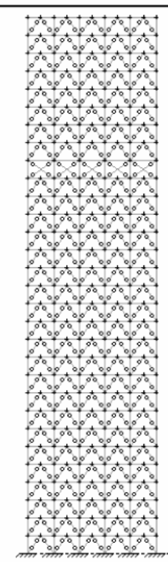
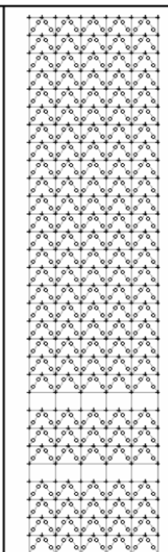
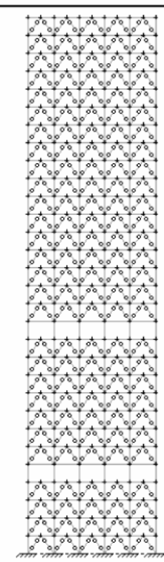
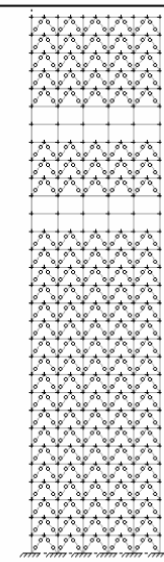
Best Designs

Best design concepts developed by totalistic 2D CA rules with Moore neighborhood are shown in Table 50. The top row presents the 6 best design concepts generated with the radius of the local neighborhood equal to 1 while the bottom row shows the 6 best concepts produced with the radius equal to 2. The total weight of the best design concept was equal to 458,274 lbs. for both lengths of the radius of the local neighborhood. It was significantly better (more than 78,000 lbs.) than the best design concept produced by a standard 2D CA rule with Moore neighborhood. The overall median values for both radii were, however, larger than the overall median value obtained for standard 2D CA rules. They were equal to 581,002 lbs. and 615,461 lbs. for the radius equal to 1 and 2, respectively.

Structural Shaping Patterns

All 12 design concepts shown in Table 50 exhibit the fully-braced structural shaping pattern consisting of K bracings. The two best design concepts, one for each radius, exhibit a fully developed pattern with all cells occupied by K bracings. The remaining 10 design concepts display some variations of the fully braced pattern. The influence of the increased radius of the local neighborhood on the generated patterns can be observed in the bottom row of Table 50. Specifically, the disruptions of the uniform pattern of K bracings by either simple X bracings or no bracings spread across the entire stories. This is not the case with the patterns generated by totalistic 2D CA rules with the radius equal to 1 (see the top row of Table 50). Here, the disruptions are localized to 2, or utmost 3, cells within a single story.

Table 50. Best design concepts of wind bracing systems developed by totalistic 2D CA rules with Moore neighborhood and the radius of the local neighborhood equal to 1 (the top row) and 2 (the bottom row)

RADIUS = 1					
					
458,274 4.8721	459,200 4.9487	468,223 4.9318	474,481 4.8861	477,755 4.8761	477,892 4.9704
RADIUS = 2					
					
458,274 4.8721	462,371 5.2316	462,518 4.9436	472,110 5.2578	473,241 5.2631	477,086 6.7184

Von Neumann Neighborhood

When von Neumann neighborhood was employed instead of Moore neighborhood, the size of the totalistic 2D CA rule space was even smaller and equal to $3^{5 \cdot 3-4} = 3^{11} = 177{,}147$ when the radius was equal to 1 and $3^{13 \cdot 3-12} = 3^{27} = 7.6 \cdot 10^{12}$ when the radius was equal to 2. Thus, each totalistic 2D CA rule with 3 possible cell values and von Neumann neighborhood could be represented as a string of 11 digits and 27 digits, respectively.

Best Designs

Best design concepts developed by totalistic 2D CA rules with von Neumann neighborhood are shown in Table 51. Similarly as before, the top row contains the best design concepts generated by the design rules with the radius equal to 1 while the bottom row shows the best designs produced with the radius equal to 2. The best design concepts developed in both cases were identical and their total weight was equal to 458,274 lbs. These concepts were also the same as the best concepts produced by totalistic 2D CA rules with Moore neighborhood (see Table 50).

The best concepts shown in Table 51 were significantly (more than 78,000 lbs. or 14 percent) better than the best design concepts produced by standard 2D CA rules with von Neumann neighborhood (see Table 46). The overall median total weights of the two samples of design concepts generated with the radii 1 and 2 were equal to 574,208 lbs. and 576,644 lbs., respectively. They were slightly larger (by 2,000-4,000 lbs.) than the overall median obtained for standard 2D CA rules with von Neumann neighborhood.

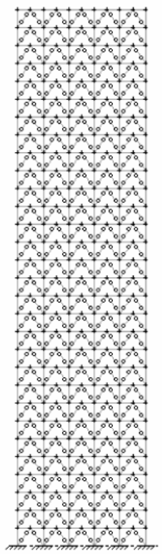
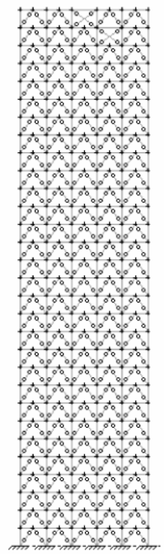
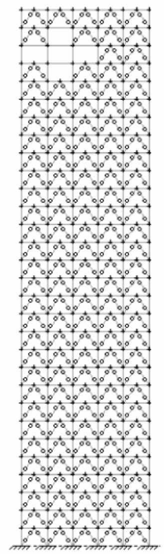
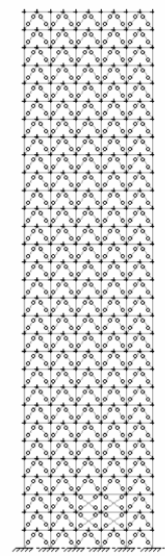
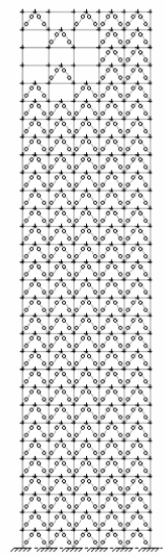
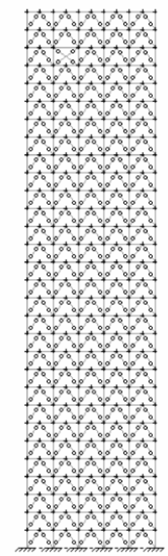
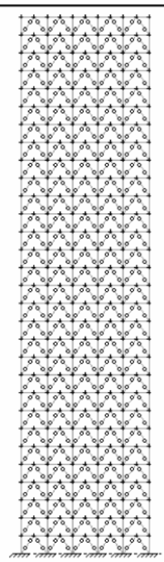
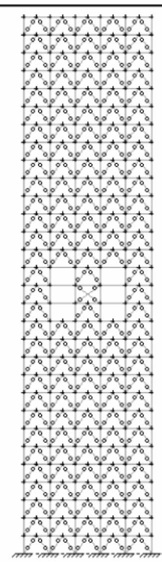
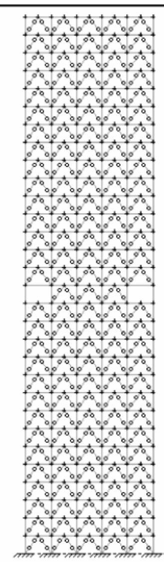
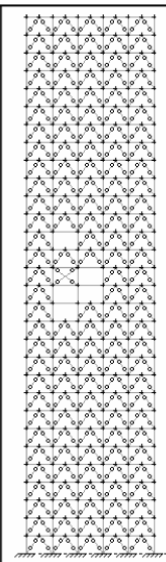
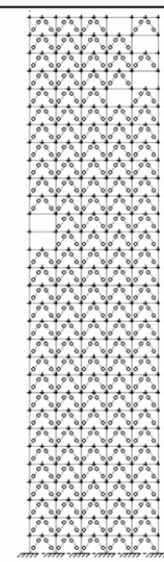
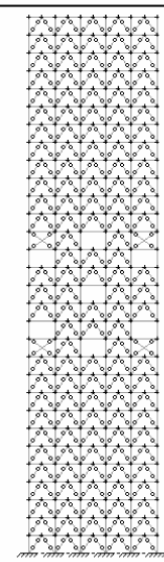
Structural Shaping Patterns

All design concepts shown in Table 51 exhibit variations of the fully-braced pattern composed of K bracings. The best design concepts had a fully developed pattern consisting exclusively of K bracings. The remaining design concepts presented in Table 51 contain some localized disruptions of the pattern in which K bracings are replaced by simple X bracings or no bracing. An interesting pattern was formed in the central part of the 6^{th} best design produced by a totalistic 2D CA rule with the radius equal to 2 (see the bottom row and sixth column of Table 51). Here, an emergent 'circular' pattern, whose width is equal to 2 stories/bays, is surrounded by simple X bracings located on the diagonals and no bracings located in the horizontal/vertical directions. The entire structural shaping pattern is symmetric.

Many other interesting structural shaping patterns have been identified during the process of iteration of totalistic 2D CA rules with von Neumann neighborhood. Figure 63 shows the process of iteration of the design rule 7366334203861 applied to a random configuration of wind bracing elements. The design rule produced more than 30 different design concepts, some of good performance, until it reached the configuration of the fully developed braced pattern consisting of K bracings.

During the process of iteration, some unique patterns emerged, particularly in the central part of the structural system. For example, at the iteration step *t=29,* a structural shaping pattern emerges which consists of 3 qualitatively diverse parts. The top and bottom parts contain the fully braced pattern consisting of K bracings. The situation is different in the central part where a pattern in the form of letter 8 emerges. It consists mostly of simple X bracings located in the central bays and stories.

Table 51. Best design concepts of wind bracing systems developed by totalistic 2D CA rules with von Neumann neighborhood and the radius of the local neighborhood equal to 1 (the top row) and 2 (the bottom row)

RADIUS = 1					
					
458,274 4.8721	458,797 4.9546	460,768 5.1066	462,009 4.8853	464,866 5.1335	465,867 4.8598
RADIUS = 2					
					
458,274 4.8721	461,826 5.0442	468,664 4.8200	470,330 5.0184	477,910 4.8389	479,479 4.9392

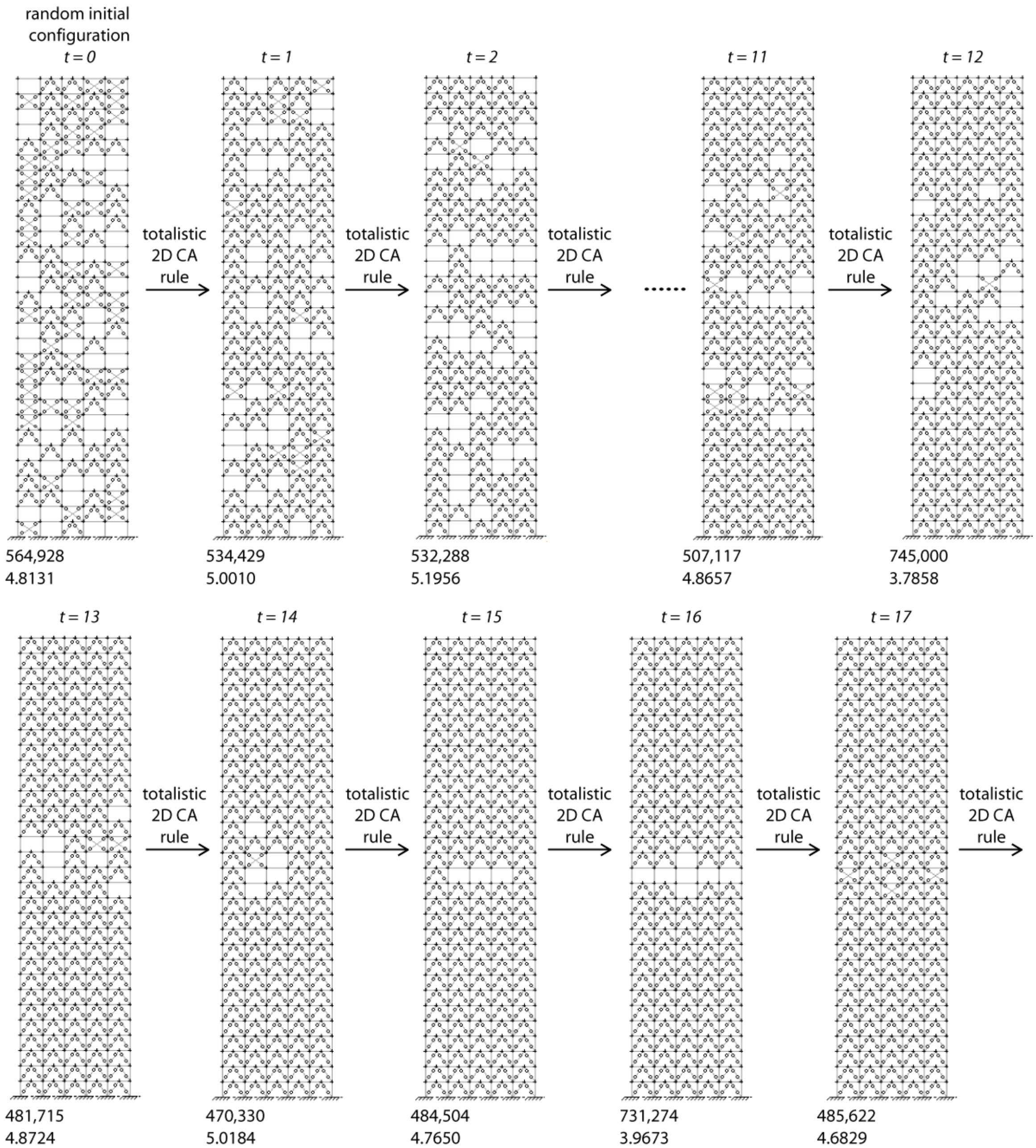


Figure 63. Process of iteration of a totalistic 2D CA rule with von Neumann neighborhood and the radius of the local neighborhood equal to 2

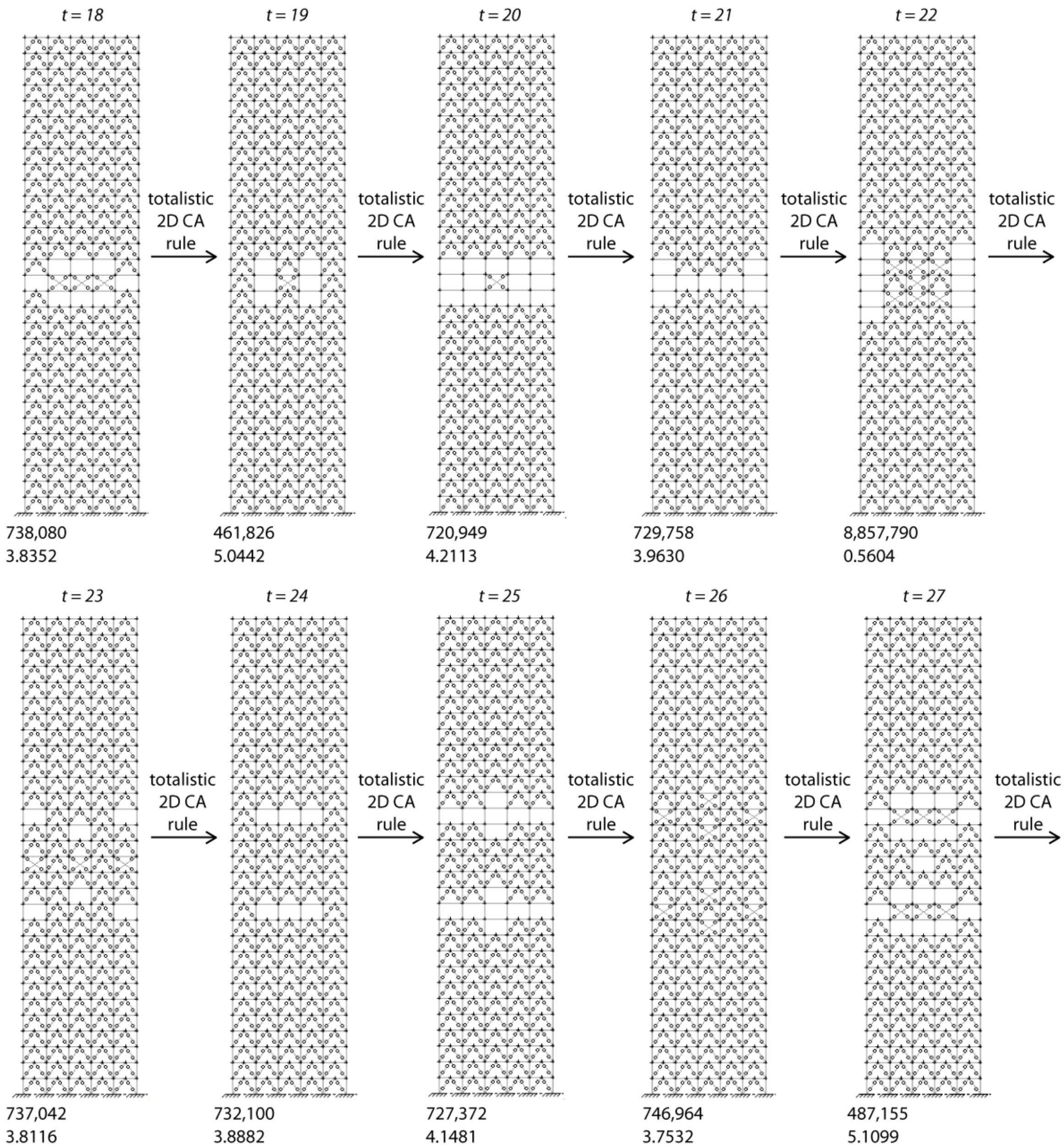


Figure 63 cont. Process of iteration of a totalistic 2D CA rule with von Neumann neighborhood and the radius of the local neighborhood equal to 2

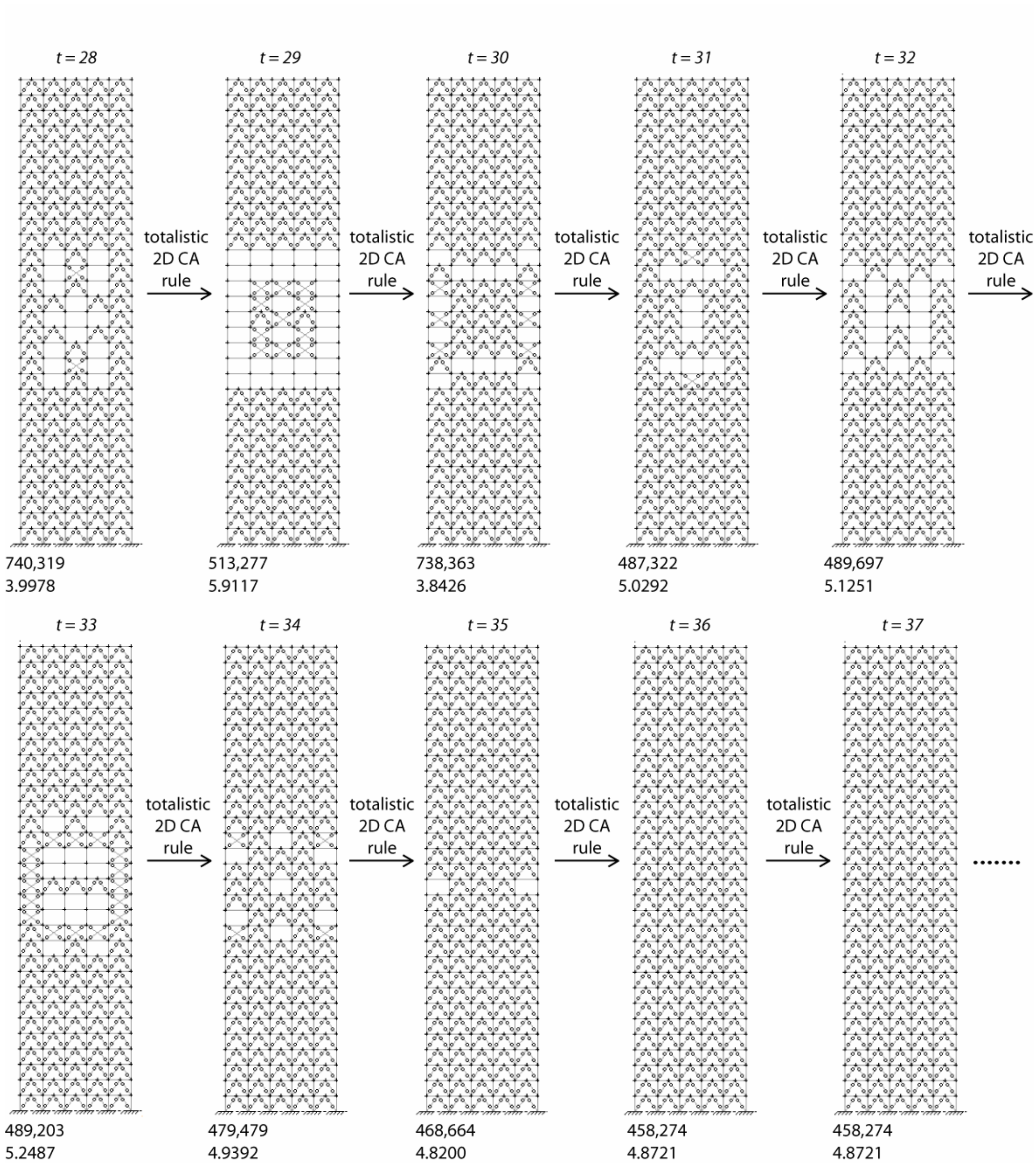


Figure 63 cont. Process of iteration of a totalistic 2D CA rule with von Neumann neighborhood and the radius of the local neighborhood equal to 2

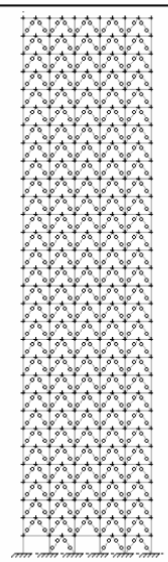
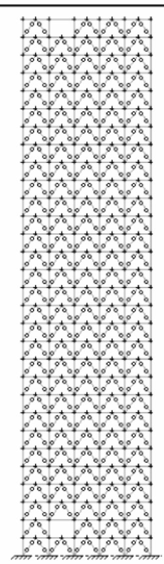
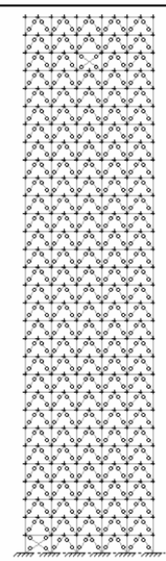
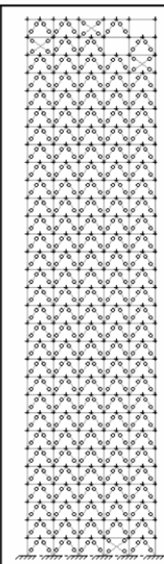
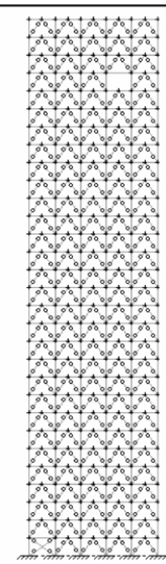
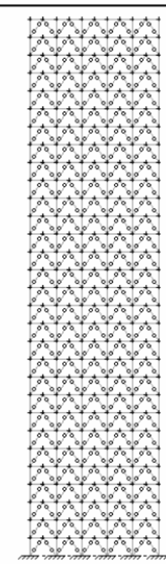
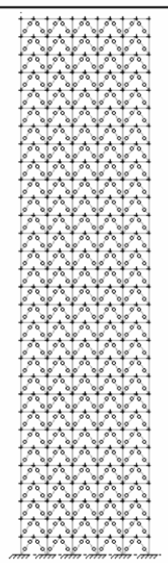
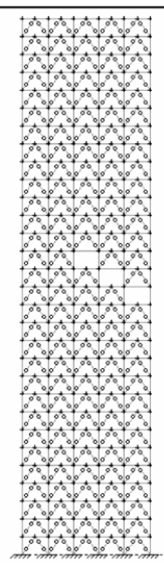
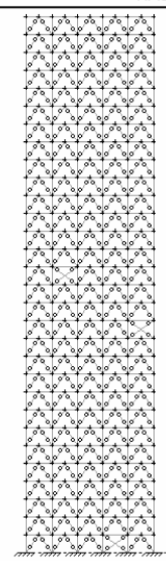
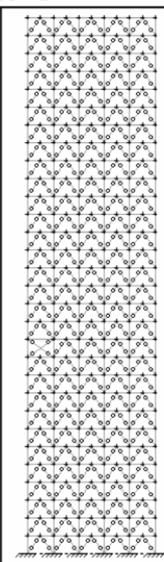
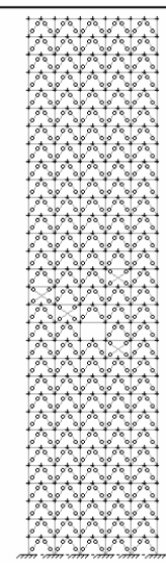
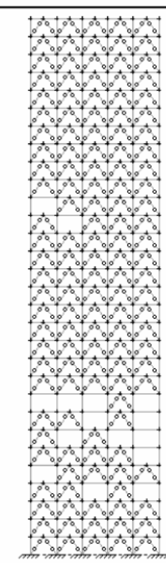
Diagonal Neighborhood

In yet another group of experiments, I investigated totalistic 2D CA with 3 cell values and the diagonal neighborhood. They formed the design rule space of size $3^{5\cdot3-4} = 3^{11} = 177{,}147$ when the radius was equal to 1 and $3^{9\cdot3-8} = 3^{19} = 1.1\cdot10^{9}$ when the radius equaled 2. Thus, totalistic 2D CA rules with 3 cell values and the diagonal neighborhood were represented by strings of 11 (radius = 1) and 19 digits (radius = 2).

Best Designs

Best design concepts developed by totalistic 2D CA rules with the diagonal neighborhood are shown in Table 52.

Table 52. Best design concepts of wind bracing systems developed by totalistic 2D CA rules with the diagonal neighborhood and the radius of the local neighborhood equal to 1 (the top row) and 2 (the bottom row)

RADIUS = 1					
					
450,209 4.9810	454,617 4.9542	456,494 4.9764	457,023 5.0018	457,595 4.9711	458,274 4.8721
RADIUS = 2					
					
458,274 4.8721	462,180 4.9207	468,769 4.9525	469,053 4.7945	472,193 4.8742	474,906 5.0891

The best design concept generated by 2D CA rules with the diagonal neighborhood and the radius of 1 had the total weight equal to 450,209 lbs.. It outperformed all design concepts found so far by about 8,000 lbs. In fact, there were 4 other design concepts which had better performance than the best fully-braced design concept found previously (see the top row of Table 52). The overall median values of the total weight of the entire samples of generated design concepts were equal to 590,924 lbs. and 580,407 lbs. for the radius equal to 1 and 2, respectively. They were larger by more than 8,000 lbs. and 18,000 lbs. than the overall median value produced in the experiments with standard 2D CA rules and the diagonal neighborhood.

Structural Shaping Patterns

The influence of the shape of the local neighborhood can be observed in several design concepts shown in Table 52. For example, the 2nd and 5th design concepts have disruptions of the uniform pattern of K bracings by either simple X bracings and no bracings occurring in the diagonal directions (see the bottom row of Table 52).

North-South Neighborhood

The north-south neighborhood and the east-west neighborhood defined the totalistic 2D CA rule spaces with smallest sizes. When the design rules based on totalistic 2D CA rules with north-south neighborhood and 3 cell values were used, then the size of the rule space was equal to $3^{3\cdot 3-2} = 3^7 = 2{,}187$ (the radius of 1) and $3^{5\cdot 3-4} = 3^{11} = 177{,}147$ (the radius of 2). In this case, each totalistic 2D CA rule was represented by a string of 7 and 11 digits, respectively.

Best Designs

The best design concepts developed by totalistic 2D CA rules with the north-south neighborhood are shown in Table 53. Here, the design concept produced by a totalistic 2D CA rule with the radius equal to 2 (the bottom row) is slightly better than the design concept produced with the radius equal to 1. It has the total weight of 458,274 lbs. and exhibits the previously identified fully-braced pattern consisting of K bracings.

The overall median values obtained for totalistic 2D CA rules with the radius equal to 1 and 2 were equal to 578,711 lbs. and 590,746 lbs., respectively, and were again larger and the value obtained for the corresponding standard 2D CA rule.

Structural Shaping Patterns

All the best design concepts presented in Table 53 exhibit various variations of the fully braced pattern. The influence of the shape of the local neighborhood can also be identified here because the disruptions of the uniform pattern mostly occur along vertical directions.

East-West Neighborhood

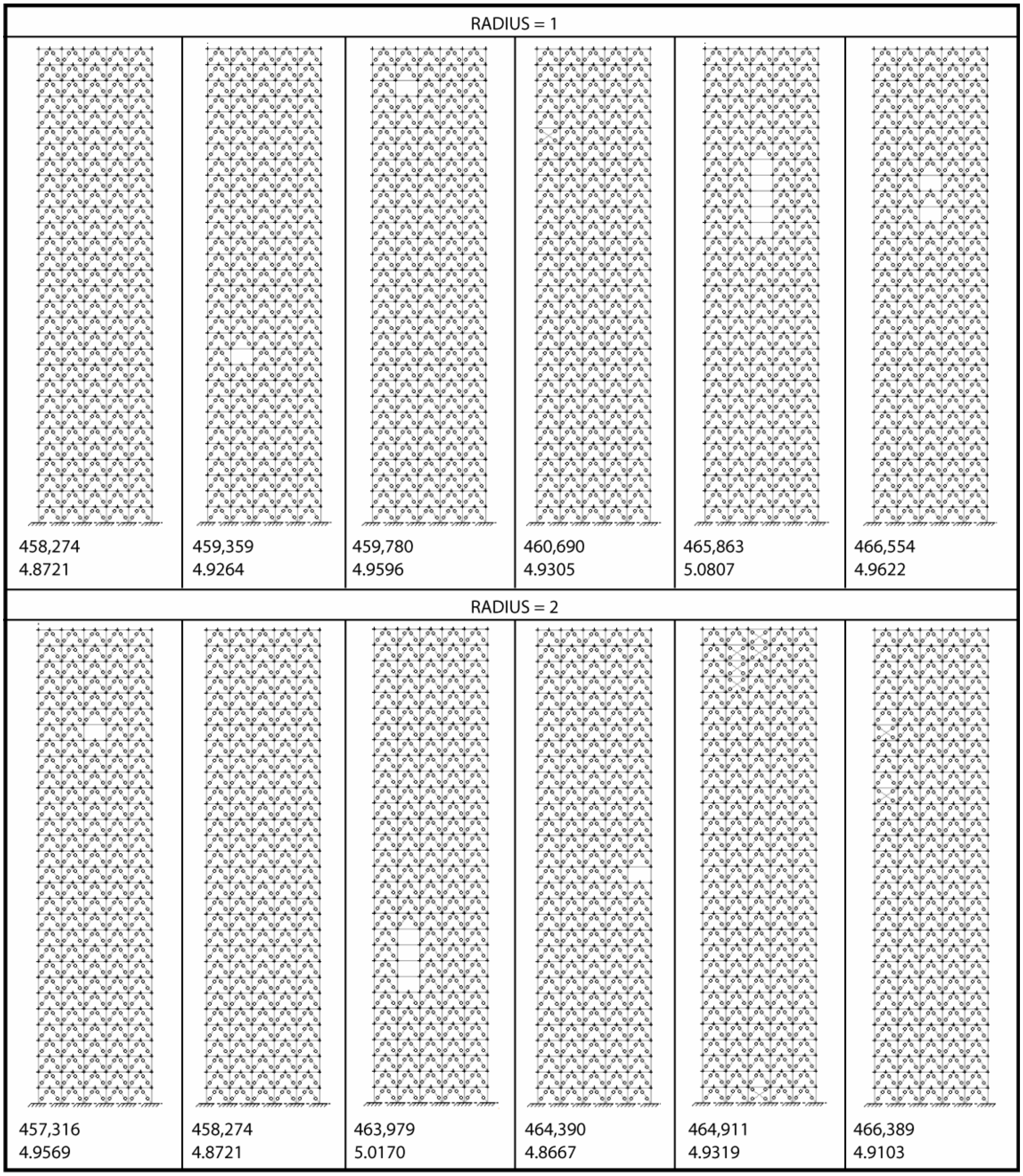
The final group of design experiments with totalistic 2D CA rules involved the east-west neighborhood. Here, the sizes of the rule spaces were exactly the same as in the case of the north-south neighborhood (see the previous subsection).

Best Designs

Table 54 shows the best design concepts developed by totalistic 2D CA rules with the east-west neighborhood. Here, the best design concept of a wind bracing system found so far has been identified. Its total weight was equal to 449,776 lbs. It exhibits the same fully-braced pattern consisting of K bracings but with a different configuration of the first story.

The overall median values obtained for the radii of 1 and 2 were equal to 606,918 lbs. and 622,112 lbs., respectively. They were significantly larger (by 26,000 lbs. and 42,000 lbs.) than the overall median value obtained for the corresponding standard 2D CA rule.

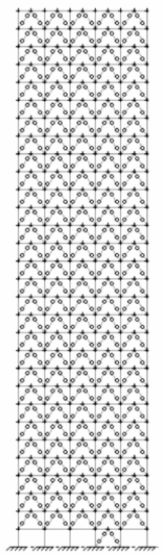
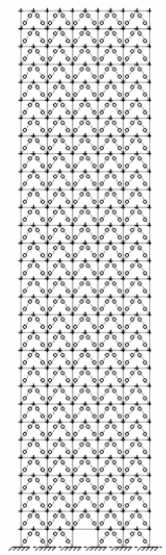
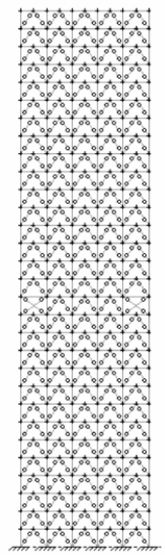
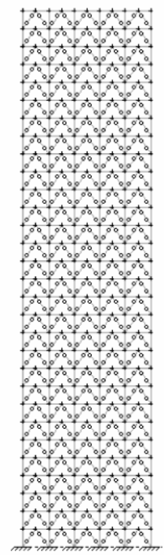
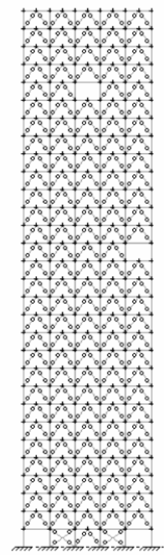
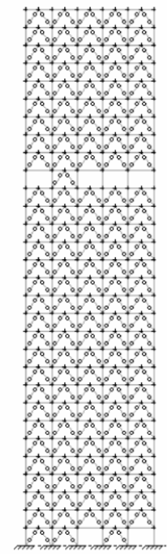
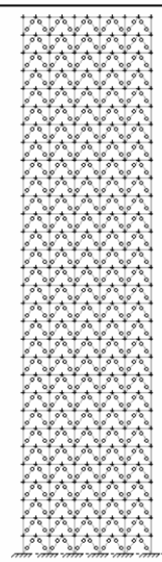
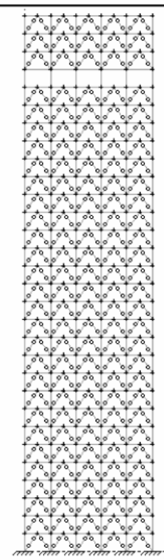
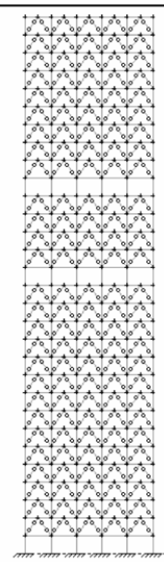
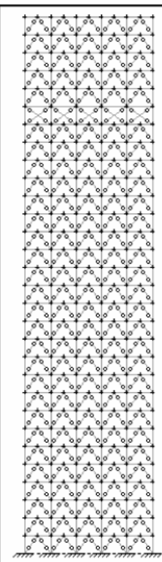
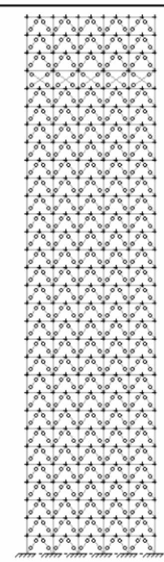
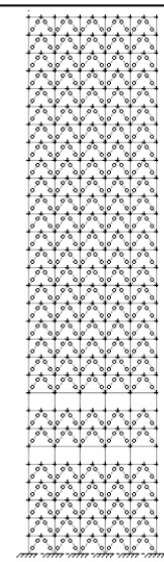
Table 53. Best design concepts of wind bracing systems developed by totalistic 2D CA rules with the north-south neighborhood and the radius of the local neighborhood equal to 1 (the top row) and 2 (the bottom row)



Structural Shaping Patterns

Similarly as before, the best design concepts exhibited slight variations of the fully-braced pattern. Also, the shape of the local neighborhood had an impact on the properties of the disruptions of the uniform pattern which occur mostly along horizontal directions. The increased radius length produced longer disruptions of the uniform pattern which spanned the entire stories.

Table 54. Best design concepts of wind bracing systems developed by totalistic 2D CA rules with the east-west neighborhood and the radius of the local neighborhood equal to 1 (the top row) and 2 (the bottom row)

RADIUS = 1					
 449,776 4.9973	 450,096 4.9893	 457,539 4.9669	 458,274 4.8721	 469,023 4.9604	 470,441 4.8842
RADIUS = 2					
 458,274 4.8721	 460,708 5.3231	 465,994 5.5940	 468,313 4.8580	 472,173 4.8486	 473,558 5.1866

Summary

The results reported in this section describe experimental studies of generative representations based on totalistic 2D CAs. As it was the case with generative representations based on standard 2D CAs, they consist of a design rule and a design embryo in a form of a 2D configuration of wind bracing elements. The design rule, however, was based on a totalistic 2D CA rather than on a standard 2D CA. This resulted in a significant reduction of the size of the rule space, in some cases by several orders of magnitude.

Similarly as in the previous section, this section was divided into five subsections which described results of the experiments obtained with 5 different shapes of the local neighborhood. Figure 64 compares the best design concepts generated by totalistic 2D CA rules with 5 different shapes and 2 radii of the local neighborhood. It also relates these results to the best design concepts produced using standard 2D CA rules discussed in the previous section.

Figure 64 clearly shows that there are no longer significant differences among the results produced by Moore, von Neumann, and diagonal neighborhoods and the north-south and the east-west neighborhoods, as it was the case with standard 2D CA rules. On the contrary, all shapes of the local neighborhood produced comparable best design concepts when totalistic 2D CA rules were employed. The results reported in this section also showed that the radius of the local neighborhood does not have a large impact on the performance of the produced design concepts but it influences the structural shaping patterns which are formed during the iterative processes.

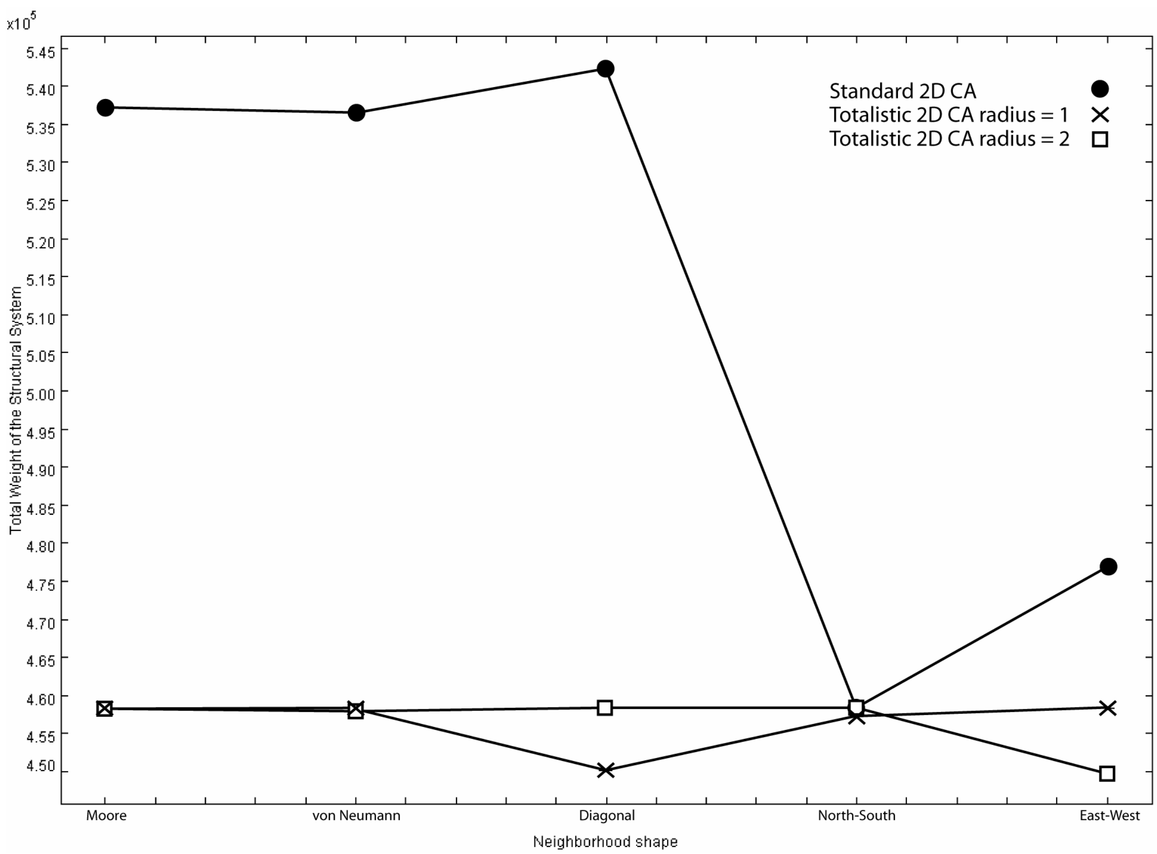


Figure 64. Comparison of the performance of the best design concepts generated by totalistic 2D CA rules with 5 shapes and two radii of the local neighborhood and their relationship to the results obtained using standard 2D CA rules

Even though the performance of totalistic 2D CA rules was similar, significant differences did exist in dynamical properties of the design processes generated by totalistic 2D CA rules with various shapes of the local neighborhood.

Figure 65 shows median total weight values of the entire samples of design concepts produced using totalistic 2D CA rules with 5 shapes and 2 radii of the local neighborhood. It also compares the median values to the corresponding values obtained in the design experiments with standard 2D CA rules.

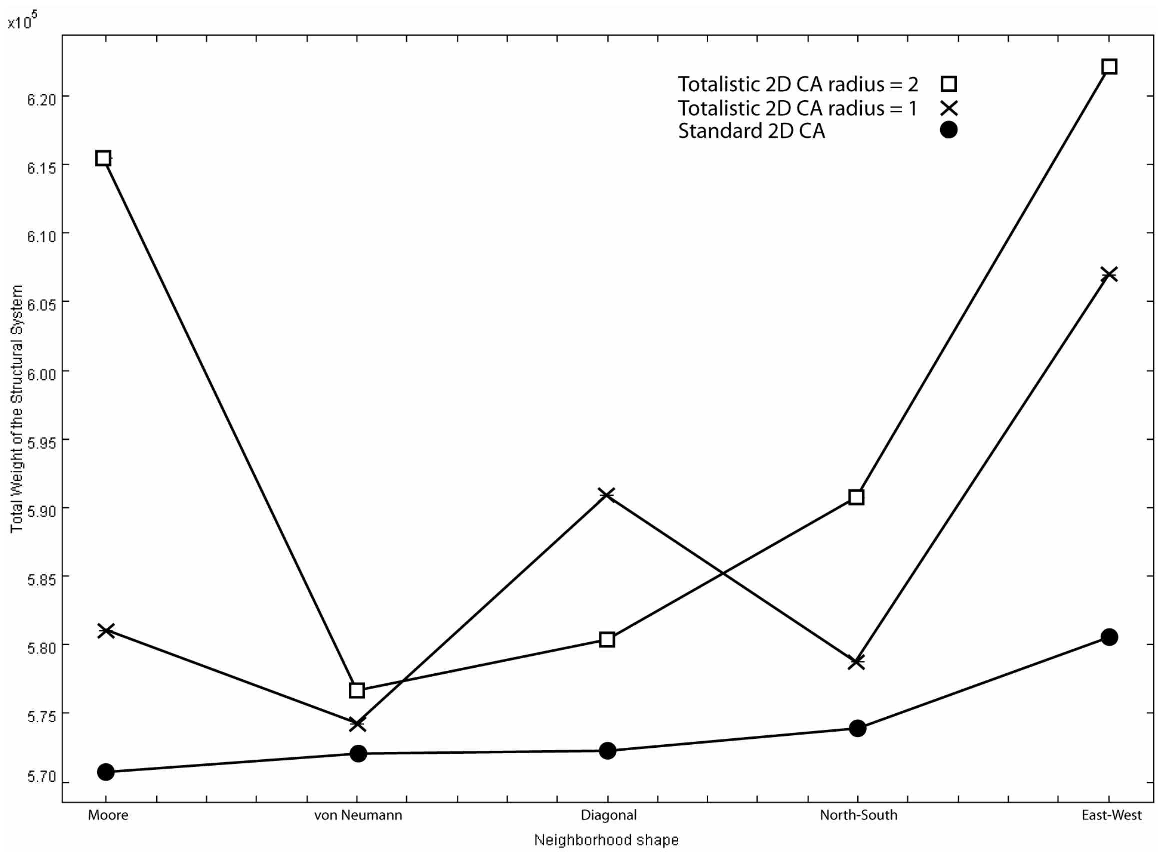


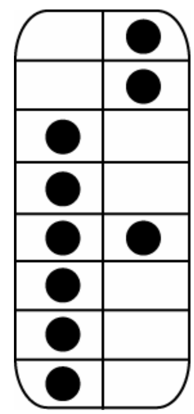
Figure 65. Comparison of the median performance of the design concepts generated by totalistic 2D CA rules with 5 shapes and two radii of the local neighborhood and their relationship to the results obtained using standard 2D CA rules

Figure 65 clearly shows that there is a tendency to generate on average heavier design concepts when we use totalistic 2D CA rules instead of standard 2D CA rules. Besides, the larger the radius of the local neighborhood in totalistic 2D CA rules the heavier the structural systems produced. The second trend occurs for all shapes of the local neighborhood with an exception of the diagonal neighborhood.

My hypothetical explanation of these results is the following: Totalistic 2D CAs significantly reduce the size of the rule spaces by taking into consideration only average values of cells in the local neighborhood. This property allows them to more quickly generate various variations of the fully braced pattern which exhibit good performance. Hence, for all 5 shapes of the local neighborhood, the design concepts with this structural shaping pattern were found. On the other hand, by the same property, totalistic 2D CA rules generate much larger changes in 2D configurations of wind bracings from iteration to iteration. This, in many cases, may produce inferior design concepts and thus, the median total weight values are larger for totalistic 2D CAs rules than for standard 2D CAs. When we increase the radius of the local neighborhood, then the changes from iteration to iteration are even larger and hence the overall median value increases again. The only exception to this trend, which is produced by the diagonal neighborhood, may be related to the emergent pattern of macro bracings which proved to correspond to design concepts of good performance. In this case, the increased radius of the local neighborhood may provide additional spatial information in the diagonal directions along which the macro bracing patterns are being formed. Hence, the median value obtained for totalistic 2D CAs with the radius equal to 2 is smaller (better) than the value obtained when the radius is equal to 1.

I also investigated the impact of the number of iterations of a totalistic 2D CA rule, denoted earlier by the *iteration_max* parameter, on the quality of produced design concepts. Similarly as in the previous section, there was no preferred value of this parameter which corresponds to better design concepts. The graphs showing these results were qualitatively the same as the ones presented in Figure 62. Thus, they have been omitted here.

6.5. Cellular Automata Generating Designs of the Entire Structural Systems



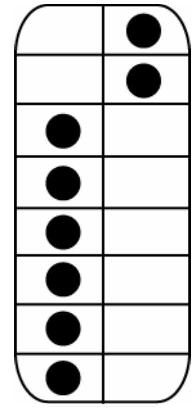
So far, I have only studied the design concept generators of wind bracing systems. In this section, I consider a more complex design problem and investigate design concept generators of the entire steel structural systems in tall buildings. They are based on multiple one-dimensional cellular automata and each 1D CA develops a separate subsystem of a steel structure, e.g. one 1D CA generates a subsystem of beams, another one a subsystem of columns, etc. A detailed description of the representations studied in this section was presented earlier in section 4.4.4.

As in the previous sections, two types of 1D CA rules have been studied. First, in section 6.5.1, I investigate the design concept generators based on multiple standard 1D CAs. Next, section 6.5.2 explores the space of design rules based on multiple totalistic 1D CAs. In both cases, I investigate only one length of the radius of the local neighborhood for each 1D CA, namely the radius equal to 1. The parameters and their values used in the experiments reported in this section are presented in Table 55.

Table 55. Parameters and their values used in the design experiments with 1D CAs generating the entire steel structural systems in tall buildings

Experimental Parameter	*Value(s)*
Number of cell values (bracings)	7
Number of cell values (beams)	2
Number of cell values (supports)	2
Radius of the local neighborhood	1
Boundary conditions	Periodic
Embryo generation mechanism	Random
Design rule search mechanism	Random
Random sample size	10,000

6.5.1. Multiple Standard 1D Cellular Automata

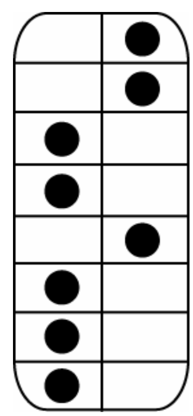


In this section, I describe the results of the experiments in which multiple standard 1D CAs were used to develop design concepts of the entire steel structural systems. The concept generators studied here used separate design embryos and separate design rules based on 1D CA to develop the subsystems of beams, wind bracings, and supports. As it is shown in Table 55, the design embryo and the design rule generating a wind bracing subsystem had 7 possible cell values, the design embryo and the design rule developing a beam subsystem had 2 possible cell values, and the design embryo encoding the supports had also 2 possible cell values. The best design concepts of the entire steel structural system produced in these experiments are shown in Table 56.

Structural Shaping Patterns

Table 56 shows that the best design concepts generated by multiple 1D CA rules exhibit several interesting structural shaping patterns. The overall best design, whose total weight was equal to 523,247 lbs., exhibits a uniform pattern consisting of V bracings and pinned beams. The third best design concept exhibits yet another type of the macro bracing pattern composed of 3 types of wind bracing elements: X bracings, diagonal bracings and V bracings. In this case, the beam subsystem is mostly composed of fixed beams with occasional occurrences of pinned beams. The macro bracing pattern can be also identified in the 6th best design concept. In this case, it emerges from a combination of X bracings, and K and V bracings. The remaining design concepts shown in Table 56 exhibit more elaborate structural shaping patterns composed of all 7 types of wind bracings elements. The majority of these design concepts have a beam subsystem composed of fixed beams only. There is, however, no predominant pattern in terms of the preferred configurations of supports. In some cases, only fixed supports were used but a vast majority of support configurations include one or more pinned supports.

6.5.2. Multiple Totalistic 1D Cellular Automata

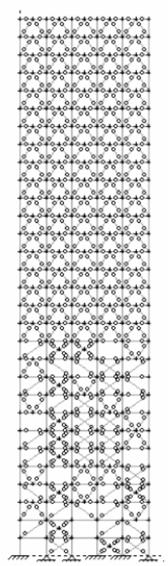
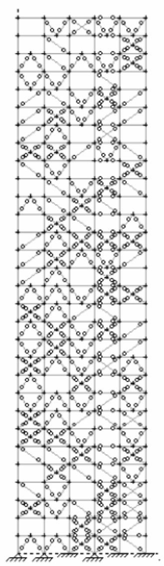
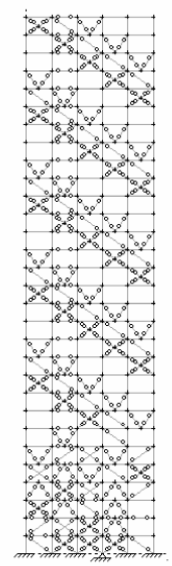
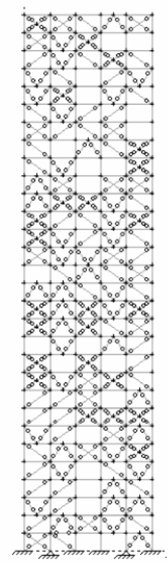
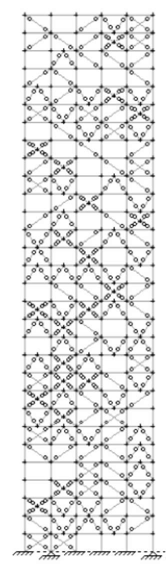
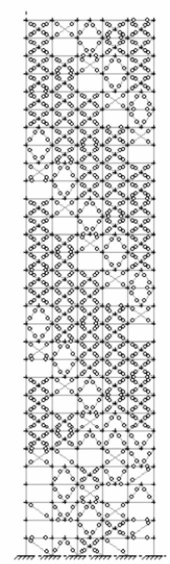
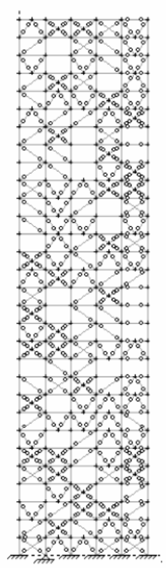
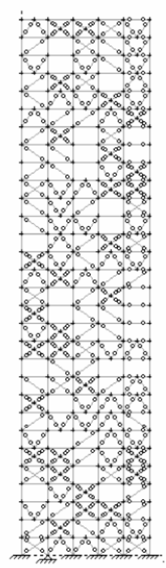
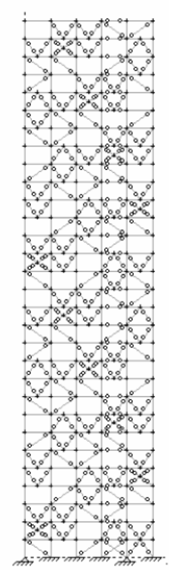
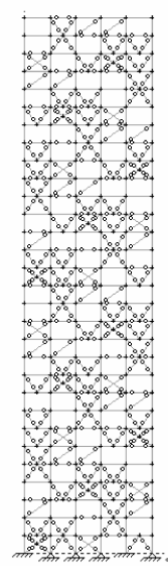
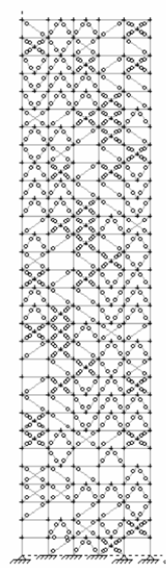
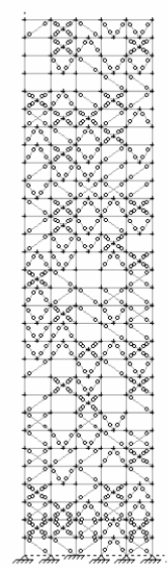


In this subsection, I describe results of the experiments in which exactly the same parameters were used as in the previous subsection (see Table 55) but with one exception: the design rules were based on totalistic 1D CA rules rather than on standard 1D CA rules. As before, the subsystems of wind bracings, beams, and supports were developed from the corresponding design embryos and design rules.

Best Designs

The best design concepts of the entire steel structural systems produced in these experiments are shown in Table 57. It is clear that there are large qualitative and quantitative differences between the design concepts produced by totalistic 1D CA rules (see Table 57) and standard 1D CA rules (see Table 56). The best design concept of the entire steel structural system in a tall building developed by totalistic 1D CA rules is more than 64,000 lbs., or 12 percent, better than the best concept produced by standard 1D CA rules.

Table 56. Best designs of the entire steel structural systems in tall buildings produced by multiple standard 1D CA rules

					
523,247 4.5228	535,713 5.1243	537,543 5.6692	538,814 4.9078	540,014 5.1149	540,391 4.8693
					
541,155 5.1890	541,165 4.9577	541,864 5.1215	542,290 6.1337	542,409 4.8599	542,529 5.1993

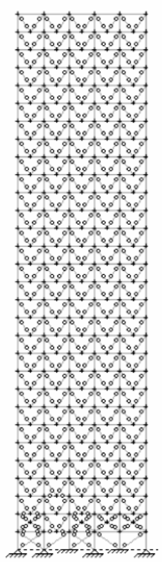
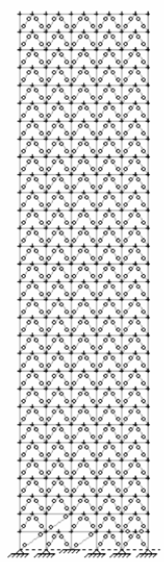
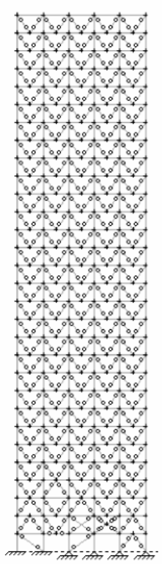
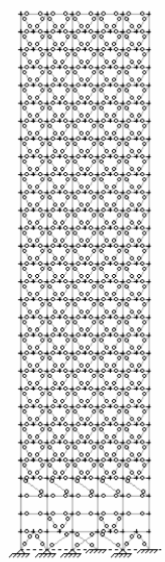
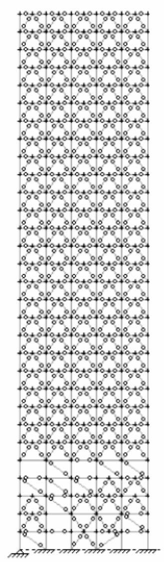
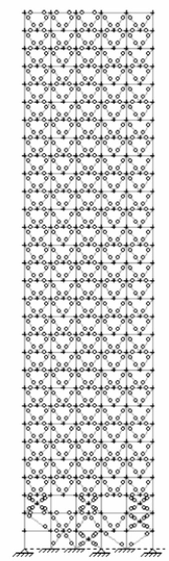
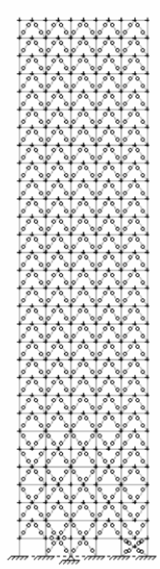
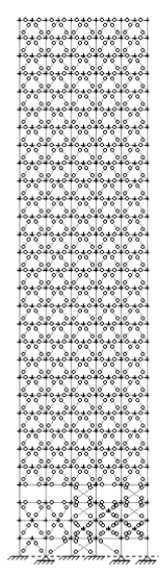
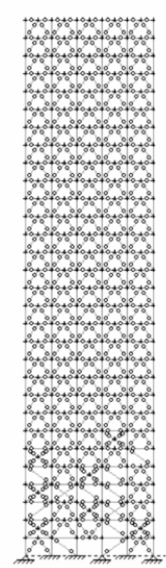
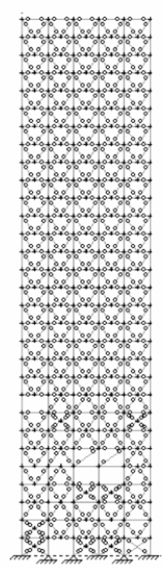
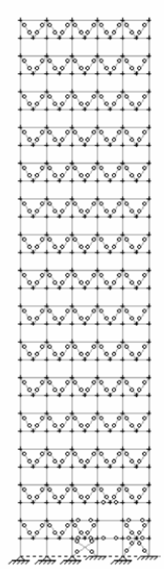
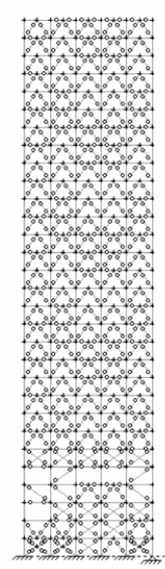
Structural Shaping Patterns

Table 57 also shows that totalistic 1D CA rules produced qualitatively different structural shaping patterns than the standard 1D CA rules. In this case, the most successful design concepts exhibited uniform patterns consisting of either V bracings or K bracings. Also, generally two types of beam subsystems were developed: composed of fixed beams only, or composed of pinned beams only. The design concepts exhibiting these patterns were of comparable performance with the total weights from about 460,000 lbs. to 480,000 lbs.

In the group of 12 best design concepts shown in Table 57, there is only one design concept which exhibits a qualitatively different structural shaping pattern, namely the 11th best design. It exhibits the pattern of horizontal trusses composed of V bracings.

Similarly as in the previous section, no clear pattern in terms of the best configuration of supports was identified. Support configurations shown in Table 57 include various combinations of fixed and pinned supports.

Table 57. Best designs of the entire steel structural systems in tall buildings produced by multiple totalistic 1D CA rules

 459,920 5.6654	 461,676 4.8454	 468,909 5.6163	 473,921 5.8685	 479,562 4.9759	 479,615 5.5117
 480,228 5.1474	 480,591 4.9270	 494,804 4.7592	 495,209 5.6509	 498,249 8.6513	 499,243 4.7361

6.5.3. Summary

In this section, I investigated design concept generators of the entire steel structural systems. They consisted of multiple design embryos and multiple design rules based on 1D CAs which generated various subsystems of the steel structure. Two types of design rules were investigated: based on standard 1D CA rules and based on totalistic 1D CA rules.

The experimental results confirm the findings reported in the previous sections in which a design of wind bracing systems was considered. Namely, standard 1D CA rules develop more interesting structural shaping patterns than totalistic 1D CA rules. The former, however, are of inferior performance (total weight) than the latter.

The design experiments have also shown that good design concepts of the entire structural systems emerge when uniform bracing patterns composed of either V bracings or K bracings are combined with uniform configurations of beams. Also, the uniform configuration of beams composed of either pinned beams or fixed beams produce comparable results.

On the other hand, no clear patterns in terms of the best configurations of supports were observed in the conducted experiments.

6.6. Summary

In this chapter, I conducted the first stage of Empirical Performance Validation of Emergent Engineering Design, as discussed in section 3.6.3. By presenting and discussing the results of the design experiments with various types of concept generation mechanisms based on cellular automata, I have attempted to build confidence in the usefulness of the generative representations component of EED. I have also shown that generative representations based on one- and two-dimensional cellular automata can produce novel design concepts of steel structural systems in tall buildings.

In the first section of this chapter, I revisited the research question 1 and the research hypothesis 1 and refined them in the context of the design problems considered in this dissertation. I also defined the criteria which were used in this dissertation to determine whether a generated design concept is novel.

In the second section of this chapter, I empirically investigated the simplest generative representations of wind bracing systems based on elementary CAs. First, I exhaustively searched the space of the design rules and applied them to the simplest configuration of the design embryo which was arbitrarily assumed. Even these very simple experimental settings produced novel structural shaping patterns of good performance. I compared the design concepts of wind bracing systems with the design concepts generated randomly and found that they perform better in terms of both the total weight of the steel structural system and its maximum horizontal displacement. I also compared the developed design concepts with the designs known from the structural engineering literature. I discovered that many traditionally known designs could be generated by the design rules based on elementary CA.

Furthermore, I investigated the impact of various representation specific parameters on the quality of generated design concepts. I found that the location of the design embryo (bottom vs. top of a steel structure) has on average no influence on the performance of produced design concepts. On the other hand, the use of nonperiodic boundary conditions may increase the total weight of steel structural systems by several percent when K bracings are used and has no impact on the quality of produced design concepts when X bracings are employed.

Next, I slightly generalized these generative representations by allowing more general configurations of the design embryo. Here, the design embryos were no longer restricted to assume the simplest possible configuration but they were generated randomly. The experimental results have shown that these more complex configurations of the design embryo produce better results and that both the space of the design embryos and the space of the design rules should be searched concurrently.

Furthermore, I demonstrated how we can incorporate some domain knowledge in the generative representation by imposing the symmetry constraint frequently used in structural design. I showed how we can constrain both components of the generative representation, i.e. the design embryo and the design rule, so that it develops symmetric design concepts. The design experiments with symmetry constraint have shown that, on average, no performance gain is achieved (in terms of the total weight of the structural system) when the symmetry constraint is imposed compared to the situation when no symmetry constraint is used and the design concepts are developed from random design embryos.

In the third section of this chapter, I studied empirically even more generalized representations based on 1D CAs. Here, the number of wind bracing types was no longer restricted to 2. In fact, in the design experiments with 1D CAs, 7 types of wind bracings elements were used. Two types of 1D CA rules were introduced and studied empirically: standard 1D CA rules and totalistic 1D CA rules. In both cases, novel design concepts of good performance were found. Also, interesting structural shaping patterns were discovered in many cases.

In the fourth section of this chapter, generative representations based on two-dimensional CAs were investigated experimentally. Also here, two types of 2D CA rules were studied: standard 2D CAs and totalistic 2D CAs. Additional parameters that needed to be specified in this type of representation included the shape of the local neighborhood and the number of iterations of the design rule.

The experiments with 2D CAs have shown that the shape of the local neighborhood has a significant influence on the performance of best design concepts only when the standard 2D CA rules are used. Totalistic 2D CA rules generated design concepts of comparable performance no matter what shape of the local neighborhood was employed. 2D CA rules produced several interesting structural shaping patterns which could not be generated by 1D CA rules due to their limitations.

The empirical studies on the impact of the number of iterations of 2D CA rules on the performance of generated design concepts have shown that there was no preferred value of this parameter which produced the best design concepts. On the contrary, for some shapes of the local neighborhood, large numbers of iterations of a 2D CA rule were preferred while for other shapes smaller values produced better results. There were, however, significant differences in the dynamical properties of the design processes when small or large numbers of iterations of the design rules were tried. The dynamics was also affected by the shape of the local neighborhood.

Finally, in the fifth section of this chapter, I investigated design concept generators of the entire steel structural systems in tall buildings. They consisted of multiple design embryos and multiple design rules based on 1D CAs which generated various subsystems of the steel structure. As in the previous sections, two types of design rules were investigated: standard 1D CA rules and totalistic 1D CA rules.

The experimental results confirmed my previous findings on the impact of type of 1D CA rules on generated structural shaping patterns and on the performance of the produced design concepts. Standard 1D CA rules developed more interesting patterns than totalistic 1D CA rules but at the same time showed inferior performance measured in terms of the total weight of the structural system.

The design experiments have also shown that very good design concepts of the entire structural systems emerge when uniform bracing patterns composed of either V bracings or K bracings are combined with uniform configurations of beams (either pinned or fixed).

7. EVOLUTIONARY OPTIMIZATION

"I have seen something else under the sun: the race is not to the swift or battle to the strong, nor does food come to the wise or wealth to the brilliant or favor to the learned; but time and chance happens to them all."

(King Solomon, Ecclesiastes 9:11)

In this chapter, I empirically investigate the evolutionary computation component of Emergent Engineering Design and its usefulness in optimizing steel structural systems in tall buildings. I describe results of a large number of design experiments which were focused strictly on design optimization issues. Due to emphasis on optimization, this chapter studies only parameterized representations of structural designs described earlier in section 4.2. The combined approach, i.e. the generative representations evolved by evolutionary algorithms, is investigated in chapter 8.

Experiments reported in this chapter have been conducted using Emergent Designer. The results presented here constitute the second stage of the Empirical Performance Validation process as discussed earlier in section 3.6.3.

Figure 66 shows how this chapter is organized. First, in an introductory section 7.1, I discuss the criteria of optimality of steel structural systems in tall buildings. I also revisit the research question 3 and the research hypothesis 3, similarly as in chapter 6, and refine them in the context of the design problems considered in this dissertation. I also provide an overview of types of experiments reported in this chapter.

In section 7.2, I describe results of design optimization experiments in which the topology of wind bracing systems was optimized using evolutionary algorithms. The experiments reported in this section are divided in two groups: experiments in which only two types of wind bracings elements were used (subsection 7.2.1) and experiments with seven types of wind bracings elements (subsection 7.2.2).

The usefulness of the evolutionary computation component of EED in optimizing more complex engineering systems, i.e. the entire steel structural systems in tall buildings, is investigated in the remainder of this chapter. Section 7.3 reports the results of single-objective optimization experiments in which the entire steel structural systems in tall buildings were optimized (minimized) with respect to the total weight. The impact of the initialization method, i.e. random initialization vs. initialization with a set of designs known from the structural engineering literature, is investigated in subsections 7.3.1 and 7.3.2, respectively.

On the other hand, section 7.4 discusses multiobjective evolutionary optimization processes in which steel structural systems were optimized with respect to two objectives, i.e. the total weight of the steel structure and its maximum horizontal displacement. This is particularly relevant for engineering design because many design problems have more than one objective. Usually, these objectives are conflicting. In this section, a simple multiobjective evolutionary optimization method was employed in which the two objectives were assigned arbitrarily defined weights and combined into a single fitness function.

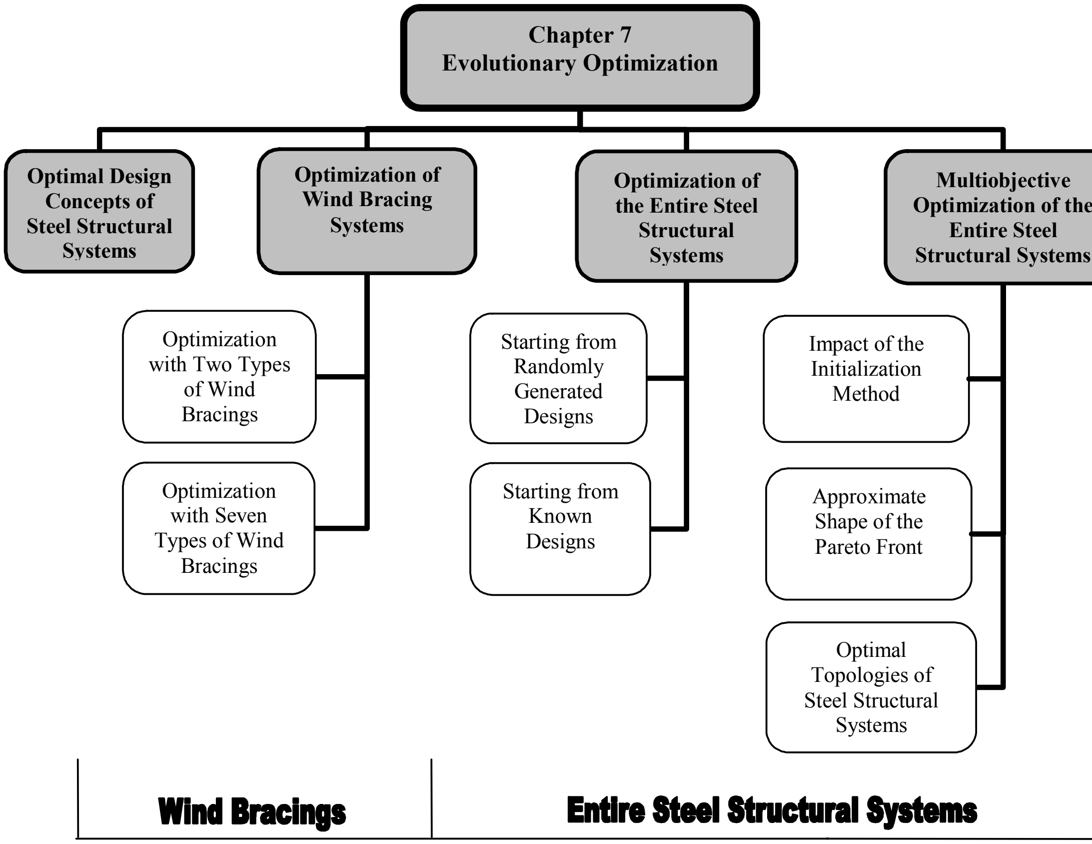


Figure 66. Organization of chapter 7

7.1. Optimal Design Concepts of Steel Structural Systems

In chapter 6, I introduced two measures of performance of steel structural systems in tall buildings: the total weight of the steel structure and its maximum horizontal displacement. The total weight of a steel structure provides a good estimate of the cost of a steel structural system while the maximum horizontal displacement estimates its stiffness. Each of the two performance measures can be used as an objective with respect to which the produced design concepts are optimized (minimized). However, the two objectives are usually conflicting. The reduction of the weight of a steel structure increases its maximum horizontal displacement (and thus reduces its stiffness) and vice versa.

The mutual interaction of the two objectives is particularly visible in steel structural systems with a large aspect ratio (see section 7.3.2). In this case, excessive reduction of the weight of a steel structural system may yield horizontal displacements that exceed provisions of the design code. Thus, the maximum horizontal displacement should be controlled when the total weight of the steel structural system is reduced as a result of the topology optimization.

In the experiments reported in the following two sections of this chapter, design concepts of wind bracing systems and the entire steel structural systems in tall buildings were optimized with respect to the total weight of the steel structure only. The maximum horizontal displacements were, however, monitored so that the design code provisions were satisfied. Later, in section 7.4, both objectives were assigned arbitrarily defined weights and multiobjective evolutionary optimization processes were investigated.

Similarly as I did in chapter 6, I can now refine the research question 3 and research hypothesis 3 in the specific context of design problems considered in this dissertation, i.e. conceptual design of steel structural systems in tall buildings.

Research Question 3 (Refined):

One of the major objectives of almost all engineering design processes is achieving optimality; what mechanisms should be used to efficiently optimize steel structural systems in tall buildings?

Research Hypothesis 3 (Refined):

Evolutionary computation provides a framework for conducting engineering design processes and efficient optimization of steel structural systems in tall buildings with respect to given objective(s).

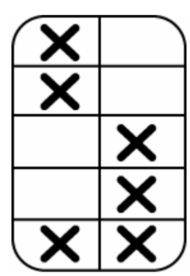
This refined hypothesis is more precise and can be tested empirically. The efficiency of evolutionary optimization processes was determined by the following two criteria:

- **Performance of the produced design concepts**
  The design concepts optimized with evolutionary algorithms were compared to the state-of-the-art designs known from the structural engineering literature and to the best designs produced by the generative representations (see chapter 6).
- **Improvements of the average best performance of the designs concepts**
  The performance of the design concepts at the end of evolutionary optimization processes was compared to the performance of initial design concepts from which the evolutionary optimization processes were started. The comparisons involved the best design concepts found in these processes as well as average performance improvements over a number of evolutionary optimization runs.

Design experiments with parameterized representations (see section 4.2) were conducted to test the research hypothesis 3. Table 58 presents the layout of design experiments reported in this chapter. All sections in this chapter are organized to follow this layout.

The experiments were divided into two major groups depending on the termination criteria used in individual evolutionary optimization runs, namely short-term experiments (up to 1,000 fitness evaluations) and long-term experiments (up to 10,000 fitness evaluations). This distinction is important from the structural design point of view because evaluations of generated design concepts are usually very expensive (more than 99% of computational time).

Extensive sensitivity analyses were conducted during the short-term experiments. They involved the following evolutionary computation parameters: mutation rates, crossover rates, sizes of parent and offspring populations, the type of the generational model, and the type of an evolutionary algorithm. Optimal settings for these parameters were sought and, once found, later utilized in the long-term experiments. The performance analysis of evolutionary optimization processes was conducted for both the short-term and the long-term experiments. It included four performance criteria presented in the bottom part of Table 58.



Similarly as I did in the previous chapter, I categorized all experiments reported here using the parameters and their values shown in Table 59. Also, an icon, similar to the one shown on the right, is placed at the beginning of each section to indicate the values of the parameters used in the experiments reported in that section.

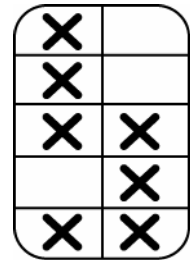
Table 58. Overview of evolutionary optimization experiments reported in this chapter

<table>
<tr><th></th><th>Short-term Experiments</th><th>Long-term Experiments</th></tr>
<tr><td rowspan="6">Sensitivity Analysis</td><td>Mutation rates</td><td rowspan="6"></td></tr>
<tr><td>Crossover rates</td></tr>
<tr><td>Size of parent population</td></tr>
<tr><td>Size of offspring population</td></tr>
<tr><td>Generational model</td></tr>
<tr><td>Evolutionary algorithm</td></tr>
<tr><td rowspan="4">Performance Analysis</td><td colspan="2">Performance comparison of best design concepts produced in evolutionary optimization processes and best designs known from the structural engineering literature</td></tr>
<tr><td colspan="2">Performance comparison of best design concepts produced in evolutionary optimization processes and best designs produced by generative representations (chapter 6)</td></tr>
<tr><td colspan="2">Performance improvement of the best design concept at the end of an evolutionary optimization process compared to the best design from an initial population</td></tr>
<tr><td colspan="2">Performance improvement of an average design concept at the end of an evolutionary optimization process compared to an average design from an initial population</td></tr>
</table>

Table 59. Parameters and their values describing the types of experiments reported in this chapter

Design problem	Wind bracings	Entire steel structural system
Fitness Measure	Single-objective	Multi-objective
Length of Evolution	Short-term	Long-term
Initialization Type	Known designs	Random
Population Size	Small	Large

7.2. Optimization of Wind Bracing Systems



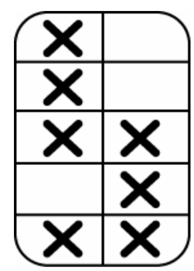
In the experiments reported in this section, the evolutionary computation component of EED was employed to optimize the topology of wind bracing systems in tall buildings. The fitness of the produced design concepts was determined by the total weight of the steel structural system (single-objective optimization) represented by these concepts. It was calculated using the first-order structural analysis.

Both the short-term and the long-term evolutionary optimization processes were conducted. By short-term evolutionary optimization processes, I understand design processes in which up to 1,000 fitness evaluations were conducted. The long-term design experiments involved significantly larger number of evaluations, even as many as 10,000. Evolutionary optimization processes were repeated several times for all combinations of parameter values and each time initialized with a different random seed value.

As mentioned earlier, an extensive evolutionary parameter search (sensitivity analysis) was conducted during the short-term optimization processes. The analysis involved the sizes of parent and offspring populations, the type of the generational model, and rates of mutation and crossover operators. The optimal combination of parameters' values found in the short-term processes was subsequently used in the long-term experiments.

The following subsections report the results of the evolutionary optimization experiments in which either only 2 types of wind bracing elements were used (as in section 6.2 in which elementary CA were studied) or all 7 types of wind bracings were employed (as in section 6.3 in which 1D CA were investigated).

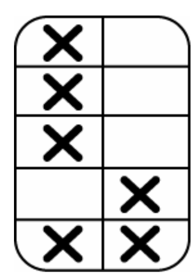
7.2.1. Optimization with Two Types of Wind Bracings



The experiments reported in this subsection involved two types of wind bracing elements. As in section 6.2, two groups of wind bracing elements were considered, each consisting of two types of wind bracings. The group No. 1 consisted of simple X bracings and no bracings (empty cells) while the group No. 2 included K bracings and no bracings (see Figure 19). The remaining types of elements of steel structural systems in tall buildings, i.e. columns, beams, and supports, were kept the same during the entire evolutionary optimization processes. Table 60 shows parameters and their values of the design problem considered in this subsection.

As discussed earlier, the design experiments with 2 types of wind bracings were divided into two groups. First, the short-term design processes were employed to conduct the sensitivity analysis involving various types of evolutionary computation parameters. Next, the optimal values of these parameters were used in the long-term evolutionary optimization processes. The results of both groups of experiments are reported in the following two subsections.

Short-term Evolutionary Optimization



In this group of experiments, the short-term evolutionary optimization processes involving two types of wind bracing elements were conducted. Evolutionary computation parameters used in the experiments reported in this subsection are presented in Table 61.

Table 60. Problem parameters and their values used in the evolutionary optimization experiments with two types of wind bracing elements

Problem Parameter	*Value(s)*
Problem type	Design of a wind bracing system in a tall building
Number of stories	30
Number of bays	5
Bay width	20 feet (6.01 m)
Story height	14 feet (4.27 m)
Distance between transverse systems	20 feet (6.01 m)
Types of bracing elements	No and Simple X, or No and K
Types of beam elements	Fixed-Fixed
Types of column elements	Fixed-Fixed
Types of supports	Fixed

Table 61. Evolutionary computation parameters and their values used in the short-term optimization experiments with two types of wind bracing elements

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies (ES), Genetic Algorithm (GA)
Generational model	Overlapping for ES(μ+λ), Nonoverlapping for ES(μ,λ) and GA
Population sizes (parent, offspring)	(1,5), (5,25), or (50,250) for ES(μ+λ) (5,25), or (50,50) for GA (5,25) for ES(μ,λ)
Selection (parent, survival)	(uniform stochastic, truncation) for ES, (fitness proportional, uniform stochastic) for GA
Mutation rate	0.025, 0.1, 0.3, or 0.5
Crossover (type, rate)	(uniform, 0), (uniform, 0.2), (uniform, 0.5)
Fitness	Total weight of the steel structure (determined by the first-order analysis)
Initialization method	Random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	1,000 fitness evaluations
Number of runs	5 in each experiment

Table 61 shows that two types of evolutionary algorithms were employed in the short-term experiments: evolution strategies (ES) and genetic algorithms (GA). Furthermore, two kinds of ES were applied, namely ES($\mu+\lambda$) and ES(μ,λ). ES($\mu+\lambda$) uses the overlapping generational model in which the survival selection acts on a combined population of parents and offspring. On the other hand, ES(μ,λ) employs the nonoverlapping generational model in which the survival selection considers only the population of offspring to choose the members of the population that will survive to the next generation (see section 2.1.1).

An extensive parameter search was conducted involving the following evolutionary parameters and their values: parent and offspring population sizes, the rate of mutation operator, and the rate of crossover operator. For all combinations of parent and offspring population sizes shown in Table 61, an exhaustive search for optimal rates of mutation and crossover was conducted. In each case, 12 combinations of mutation and crossover rates were considered, i.e. (mutation rate 0.025, crossover rate 0), (mutation rate 0.025, crossover rate 0.2), etc. The design processes were repeated 5 times for each combination of parameter values using a different value of a random seed each time.

The initial population of parents was generated randomly in every experiment reported in this section. Each design concept was represented by a fixed-length genome. The genome consisted of 150 genes (30 stories · 5 bays) with binary values. The genes represented binary attributes in which the value of 0 denoted no bracing (empty cell) and the value of 1 encoded either simple X bracing or K bracing depending on the group of wind bracing elements used in a specific design experiment.

The fitness of a design concept was determined by the total weight of the steel structural system calculated using the first-order structural analysis. Whenever an infeasible design concept was generated, it was assigned the fitness value of 0. In other words, the death penalty method was used to handle infeasible solutions (see section 2.1.4). Finally, each experiment was conducted for 1,000 fitness evaluations.

The following subsections describe the obtained results.

Optimal Rates of Mutation and Crossover Operators

Initial experiments focused on finding the optimal rates of mutation and crossover operators understood here as the rates which produced the best progress of evolutionary optimization processes. An extensive parameter search was conducted to determine the optimal rates. It involved 12 combinations of mutation and crossover rates.

The obtained results differed for various types of evolutionary algorithms. Typical results for ES are presented in Figure 67 which shows the average best-so-far fitness values and 95% confidence intervals (vertical lines) calculated using Johnson's modified t test (Johnson 1978) obtained in a series of design experiments with ES(5+25). In these experiments, the rate of uniform crossover was equal to 0.2.

A clear pattern can be identified in Figure 67 regarding the impact of the mutation rate on the fitness of produced design concepts: the lower the mutation rate the better fitness (i.e. lower because it is a minimization problem) of design concepts produced. This pattern was observed in all design experiments involving ES with various parent and offspring population sizes, and crossover rates, as it is illustrated graphically in Figure 68.

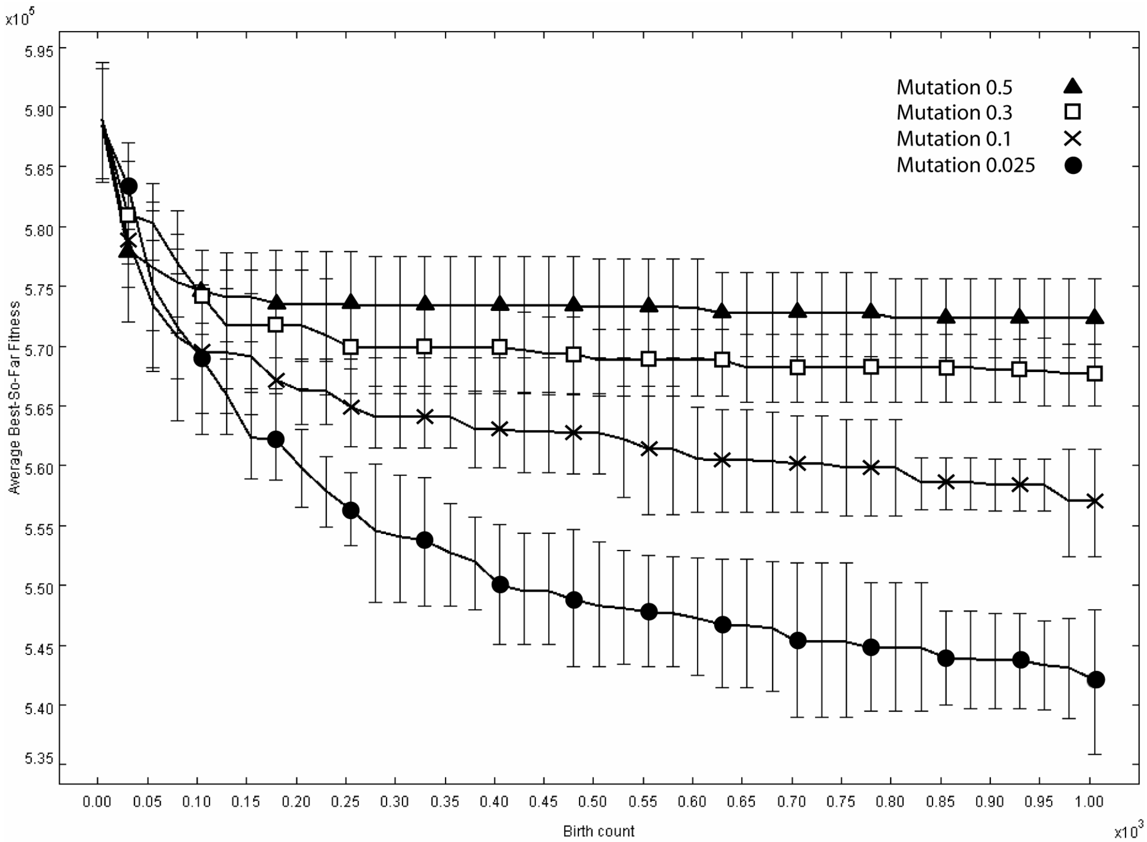


Figure 67. The influence of the mutation rate on the progress of the short-term evolutionary optimization processes when two types of wind bracing elements (no bracing and simple X bracing) were used

Figure 68 clearly shows that the best performance of ES(μ+λ) in the short-term evolutionary design processes was obtained when the mutation rate was the lowest and equal to 0.025. The same pattern was observed in the design experiments with the second group of wind bracing elements, i.e. the group consisting of no bracings and K bracings. Figure 69 shows a typical example of the impact of various mutation rates on the performance of the evolutionary optimization process. It specifically shows the results of the experiments in which ES(5,25) was used and the crossover rate was equal to 0.2.

A search for the optimal rate of the crossover operator was conducted by analyzing the results of the design experiments in which various crossover rates were used but the mutation rate was kept the same. Figure 70 presents typical results obtained in these experiments. It shows the average best-so-far fitness values and 95% confidence intervals obtained in the design experiments with ES(5+25) and 3 different rates of crossover, i.e. 0.0, 0.2, and 0.5. The mutation rate was kept the same and equal to 0.025.

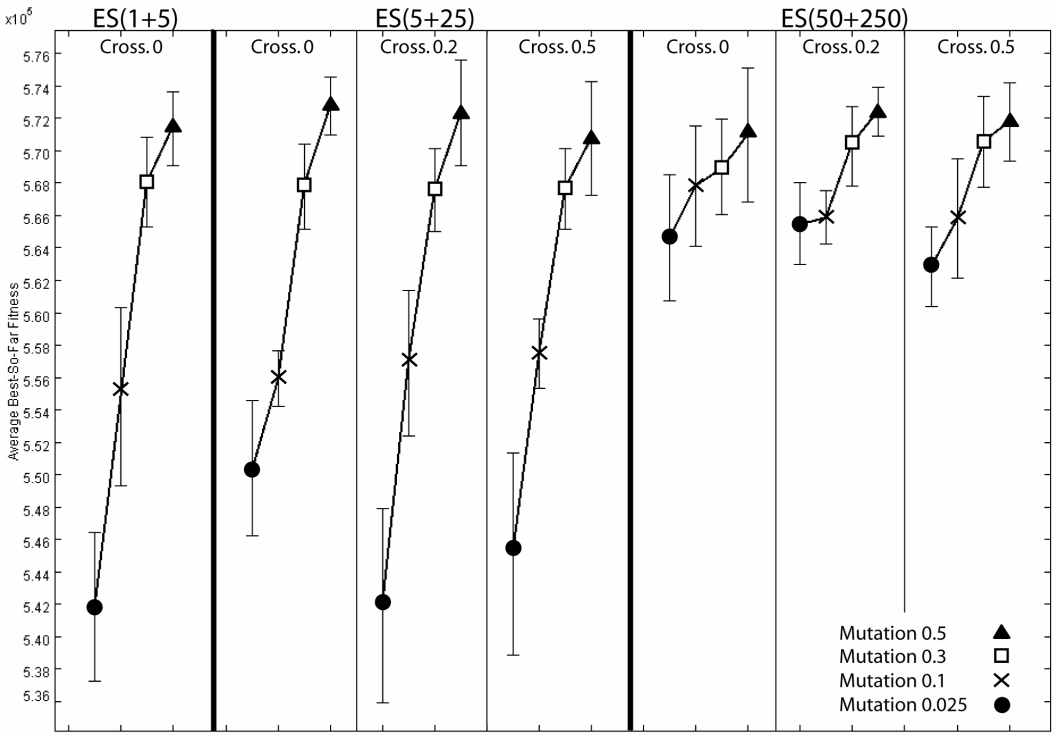


Figure 68. Average fitness values (and 95% confidence intervals) obtained after 1,000 evaluations using ES with the overlapping generational model and various parent and offspring population sizes, and mutation and crossover rates

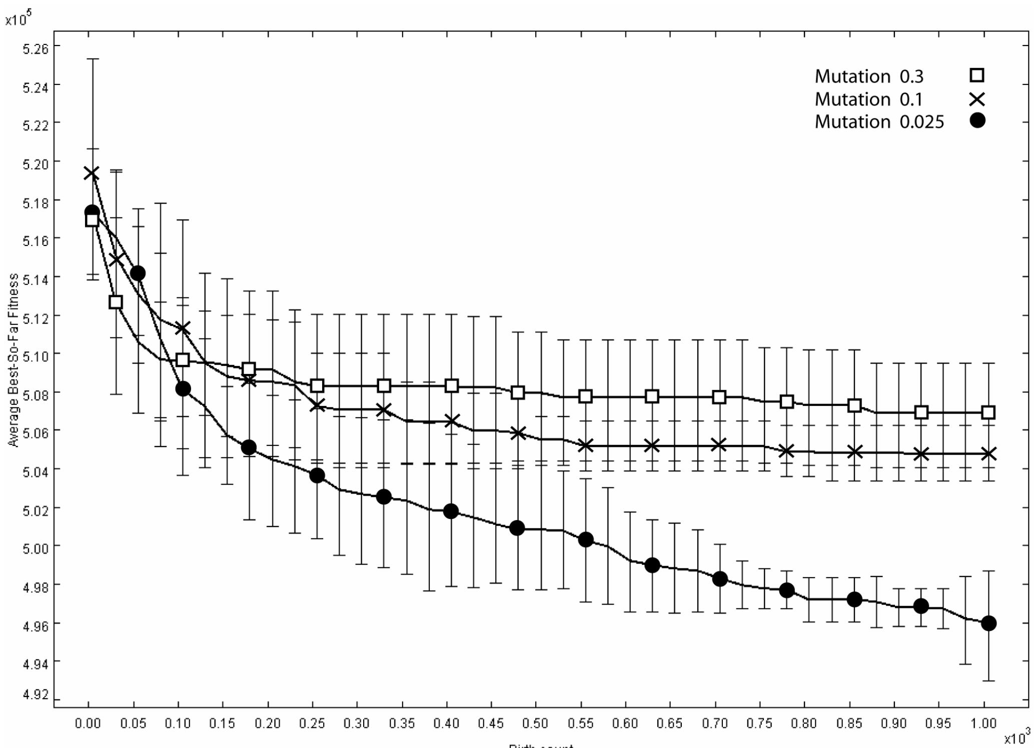


Figure 69. The influence of the mutation rate on the progress of the short-term evolutionary optimization processes when K bracings were used

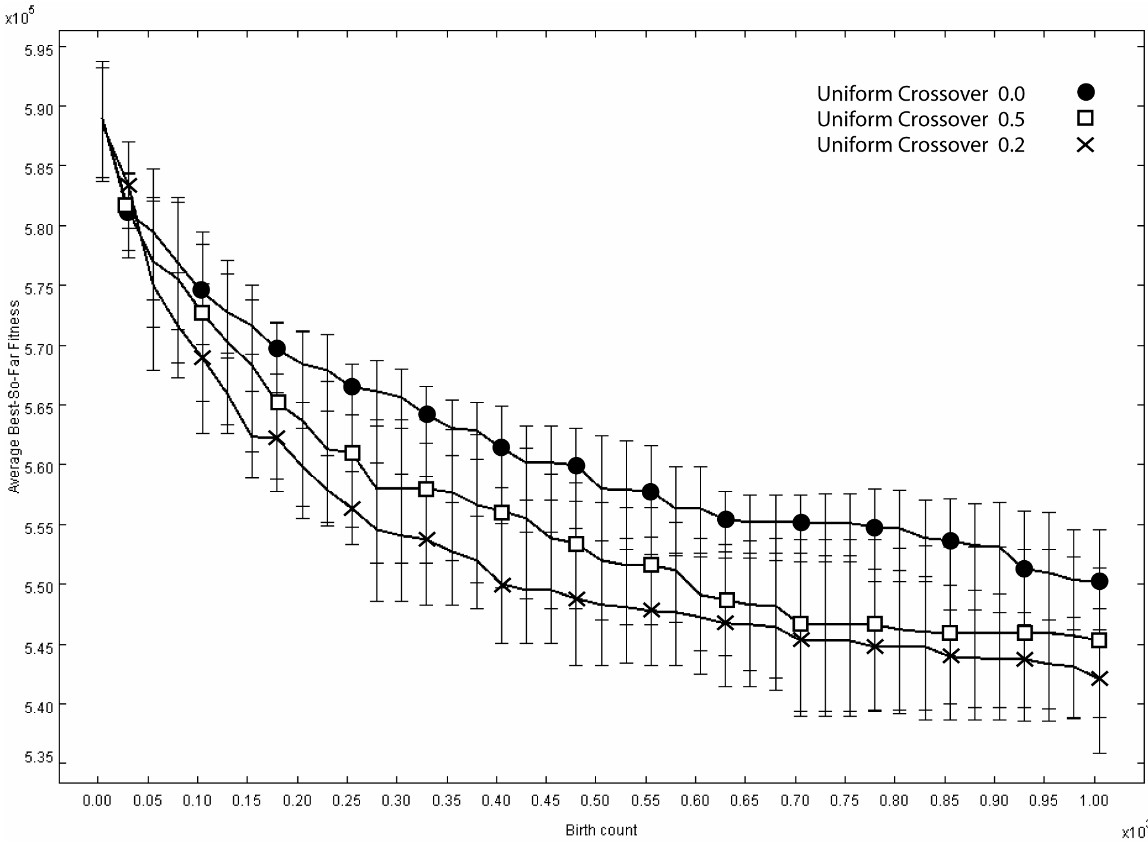


Figure 70. The influence of the crossover rate on the progress of the short-term evolutionary optimization processes when two types of wind bracing elements (no bracing and simple X bracing) were used

Figure 70 shows that various crossover rates yielded only slight differences in the fitness of produced design concepts. No clear pattern could be observed, as it was the case with the mutation operator. These observations were further confirmed by the results presented in Figure 71. It shows that there was no trend which favored specific crossover rates. On the contrary, in some cases the best results were achieved with no crossover at all and sometimes the best results were obtained when very high crossover rates are used, i.e. when the rate was equal to 0.5. Figure 71 also shows that even if there were differences among the fitness values obtained with various crossover rates, they were not significant (confidence intervals overlap in all cases). These results were consistent for both groups of wind bracing elements used in the experiments.

As I mentioned earlier, the results obtained using GAs were quite different than the ones produced by ES. Figure 72 compares the results produced by the two algorithms. Here, the graphs produced by ES(5,25) (left), using this time the nonoverlapping generational model, are compared to the graphs produced by GA(5,25) (center) and GA(50,50) (right). GAs traditionally employ the nonoverlapping generational models. Hence, the nonoverlapping generational model was used in all experiments involving GA.

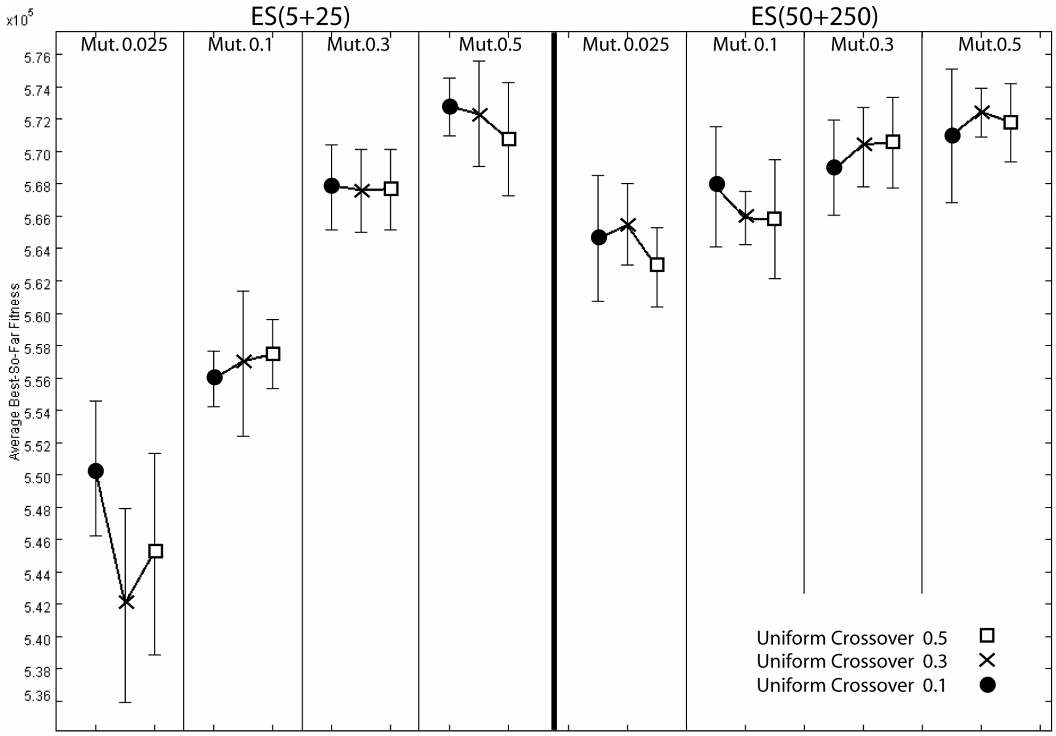


Figure 71. Average fitness values (and 95% confidence intervals) obtained after 1,000 evaluations using ES with the overlapping generational model and various parent and offspring population sizes, and mutation and crossover rates (sorted with respect to the mutation rate)

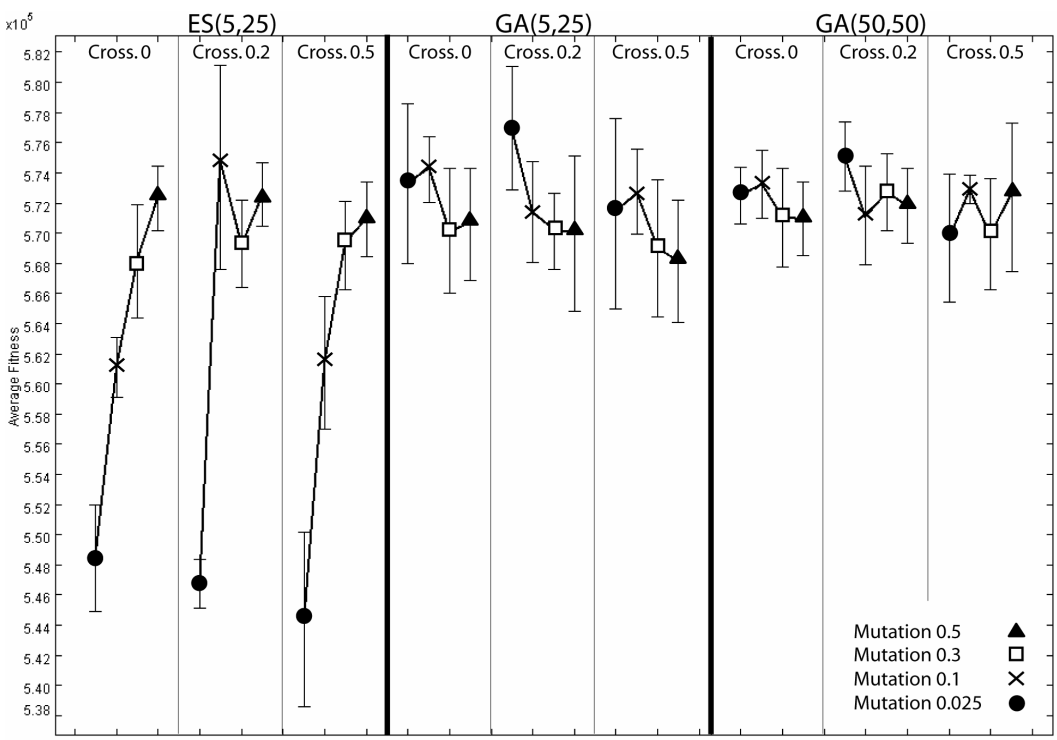


Figure 72. Average fitness values (and 95% confidence intervals) obtained after 1,000 evaluations using ES and GA with the nonoverlapping generational model and various parent and offspring population sizes, and mutation and crossover rates

It is clear that ES with the nonoverlapping generational model exhibits the same pattern as the one produced by ES with the overlapping generation model shown earlier in Figure 67. On the other hand, the results produced by GAs suggest the opposite trend: higher mutation rates produce better results, particularly when low crossover rates are used. In this case, however, the differences among the results produced by GAs with various rates of mutation are small. Finally, similarly as in the case of ES, GAs do not exhibit any clear pattern in terms of preferred crossover rates. The graph showing these results was, however, omitted.

Concluding, ES seem to produce the best results when low rates of mutation operator are used, e.g. 0.025. On the contrary, higher rates of mutation seem to be preferred by GAs but the differences in the obtained results are not as significant as in the case of ES. Neither ES nor GAs exhibit any pattern in terms of optimal crossover rates.

Optimal Sizes of Parent and Offspring Populations

The next group of experiments focused on determining the optimal sizes of populations of parents and offspring. Three different combinations of sizes of parent and offspring populations were considered for ES and two combinations for GAs.

Typical results obtained for ES are presented in Figure 73. It shows the results of the evolutionary optimization experiments in which three combinations of the parent and offspring population sizes were used, including ES(1+5), ES(5+25), and ES(50+250). Mutation and crossover rates were kept the same in all experiments shown in Figure 73 and equal to 0.025 and 0.2, respectively.

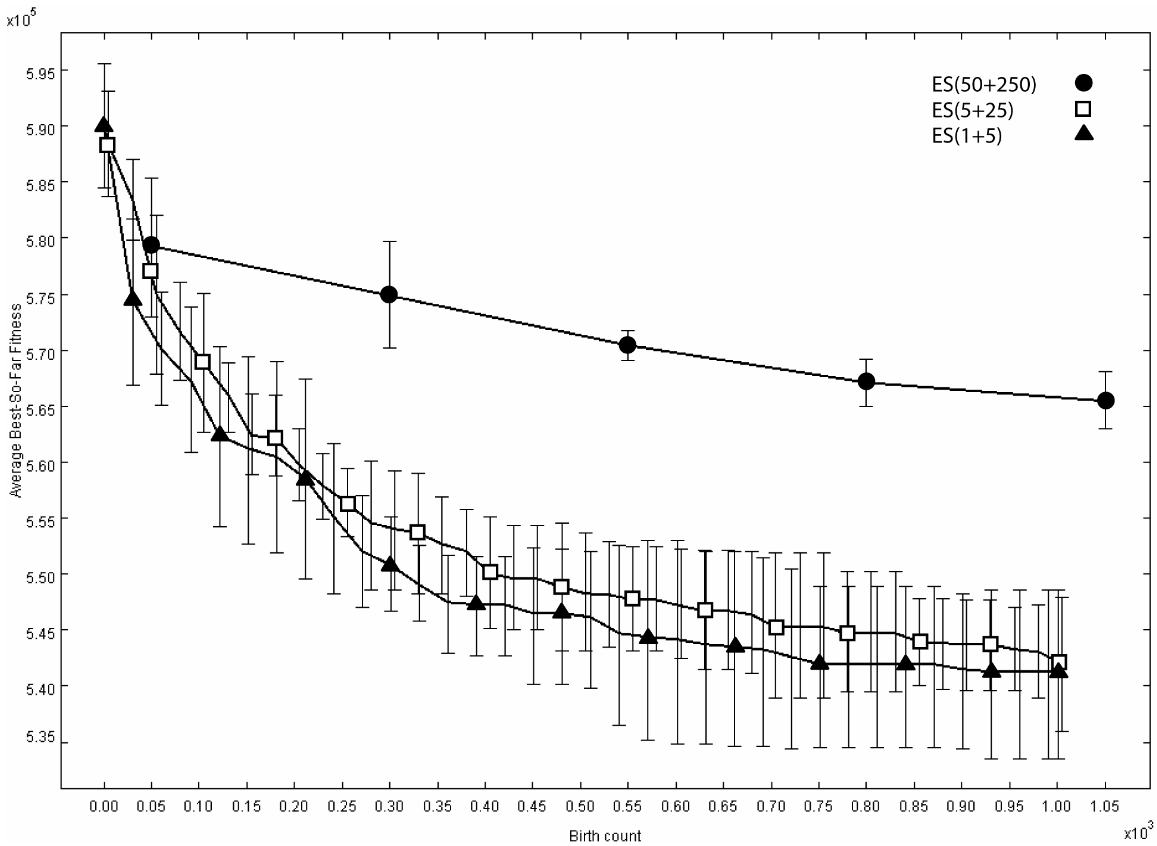


Figure 73. The influence of the sizes of parent and offspring populations on the progress of the short-term evolutionary optimization processes when ES with the overlapping generational model are used

It is clear that ES using large population sizes, i.e. ES(50+250), produced inferior results compared to the other two ES with smaller population sizes. On the other hand, it also produced the smallest variance. The other two ES with smaller population sizes achieved almost the same optimization progress in terms of the average best-so-far fitness of the produced design concepts. However, ES(1+5), i.e. the 'greedy' ES which preserves only the best individual to the next generation, exhibited much larger variance compared to ES(5+25) which preserves the top 5 individuals to the next generation. Thus, in this case parallel search conducted by ES(5+25) reduces the variance of the obtained results without decreasing the performance of the algorithm. On the other hand, when we increase the size of the populations too much, e.g. as in ES(50+250), the reduction of variance comes at a cost of a substantial decrease of the performance of the algorithm.

The outcomes were again different for GAs. In both cases, i.e. for GA(5,25) and GA(50,50), the performance of the algorithm was almost identical. Figure 74 shows typical results of the design experiments involving GA(5,25) and GA(50,50). The specific results presented in this figure were produced by the two algorithms with the same mutation and crossover rates equal to 0.3 and 0.5, respectively.

The two best-so-far curves are almost identical. The only difference between the two curves is the reduction of variance for the algorithm with larger population sizes, i.e. for GA(50,50). Similar behavior was also observed for ES.

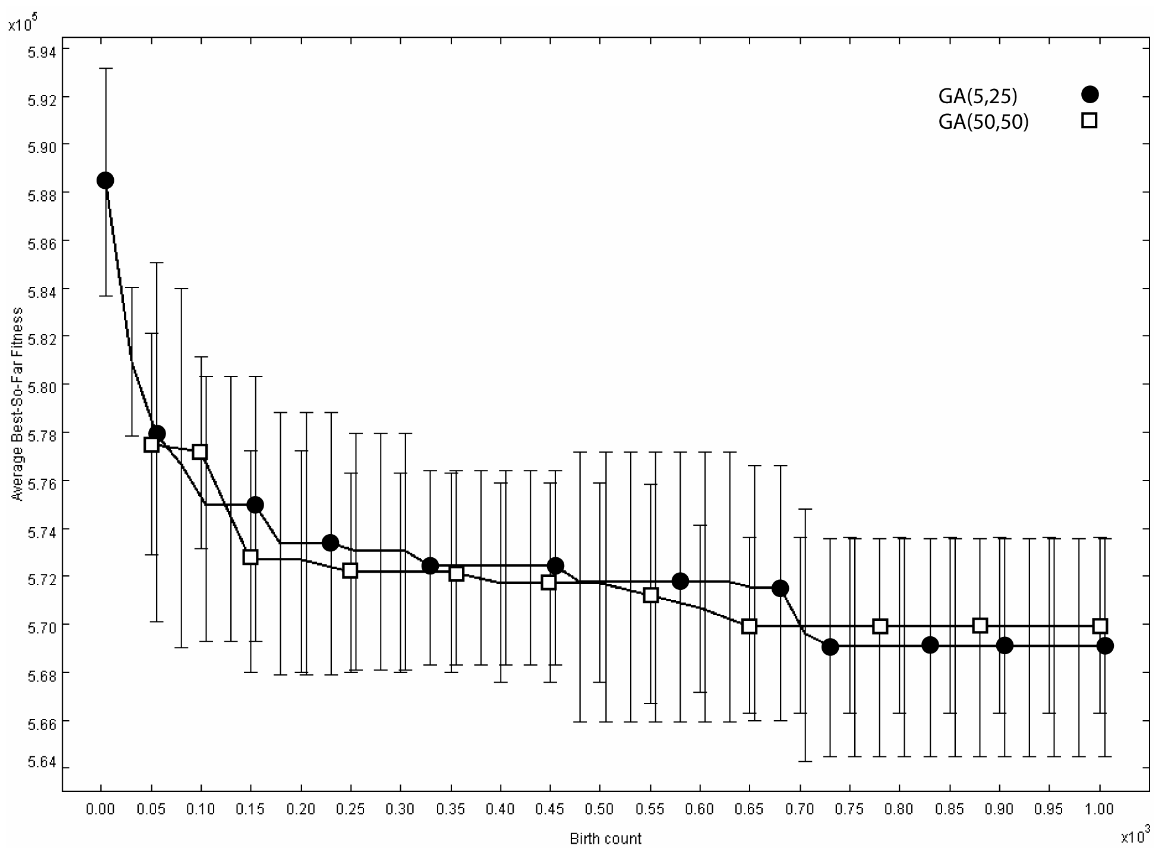


Figure 74. The influence of the sizes of parents and offspring populations on the progress of the short-term evolutionary optimization processes when GAs are used

Concluding, small population sizes seem to be preferred by ES in this problem domain. However, too small population sizes increase the variance of the obtained results. Good results in terms of both performance and variance were produced when moderate sizes of population sizes were employed, e.g. 5 in the case of the parent population and 25 in the case of the offspring population. The impact of the sizes of parent and offspring populations on the performance of GAs seems to be negligible and related only to the reduction of variance of the obtained results. It didn't influence the actual performance of the algorithm in this problem domain.

Optimal Generational Model

The influence of the type of the generational model (overlapping vs. nonoverlapping) was tested in a series of design experiments involving two kinds of ES: ES(5+25) and ES(5,25). The design experiments included a total of 24 design experiments (12 for each algorithm) utilizing all 12 combinations of mutation and crossover rates (see Table 61). Figure 75 shows typical results obtained in these experiments. Here, mutation and uniform crossover rates were equal to 0.025 and 0.2, respectively.

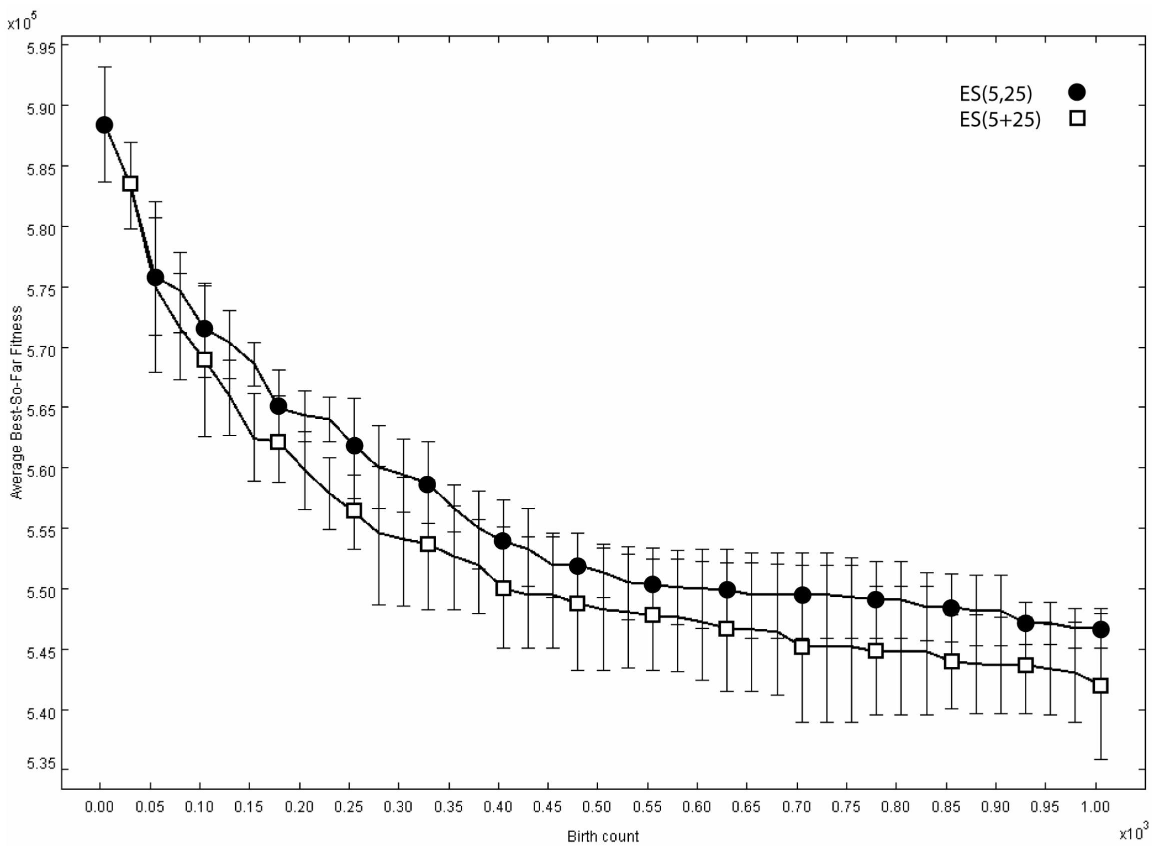


Figure 75. The influence of the type of the generational model on the progress of short-term evolutionary optimization processes

Figure 75 shows that there are no significant differences between ES(5,25) and ES(5+25). This type of behavior was observed in all conducted experiments. In several cases ES(5+25) slightly outperformed ES(5,25) (as in Figure 75) but in other cases it produced inferior results. The differences between the two generational models were, however, small both in terms of variance and fitness of the generated design concepts. Generally, it can be concluded that ES with the overlapping and nonoverlapping generational model produce comparable results in this problem domain.

Optimal Evolutionary Algorithm

As discussed earlier, short-term experiments with two types of wind bracing elements involved two types of evolutionary algorithms: ES and GAs. Sensitivity analyses were conducted for various parameters in the case of both algorithms.

Figure 76 shows a comparison of the behavior of the two algorithms optimizing a wind bracing system in a tall building. Two average best-so-far curves in the upper part of Figure 76 correspond to the best results obtained with GAs with two combinations of parents and offspring population sizes, i.e. GA(5,25) and GA(50,50). In both cases the mutation rate was equal to 0.3 and crossover rate was equal to 0.5. The results produced by GAs are compared to the average best-so-far performance produced by ES with the overlapping (ES(5+25)) and nonoverlapping (ES(5,25)) generational model. In this case, the rates of mutation and crossover were equal to 0.025 and 0.2, respectively.

Figure 76 clearly shows that ES outperformed GAs in this problem domain. The average fitness value produced by GA(5,25) after 1,000 evaluations was equal to 569,056 lbs. compared to 542,029 lbs. achieved by ES(5+25). This corresponds to almost 5% better results, on average, produced by ES. The performance improvement (see Table 58) between an average design concept produced after 1,000 fitness evaluations and an average initial parent was equal to 19,434 lbs., or 3.3%, for GA(5,25). On the other hand, for ES(5+25) these values were equal to 46,461 lbs. and 7.9%, respectively.

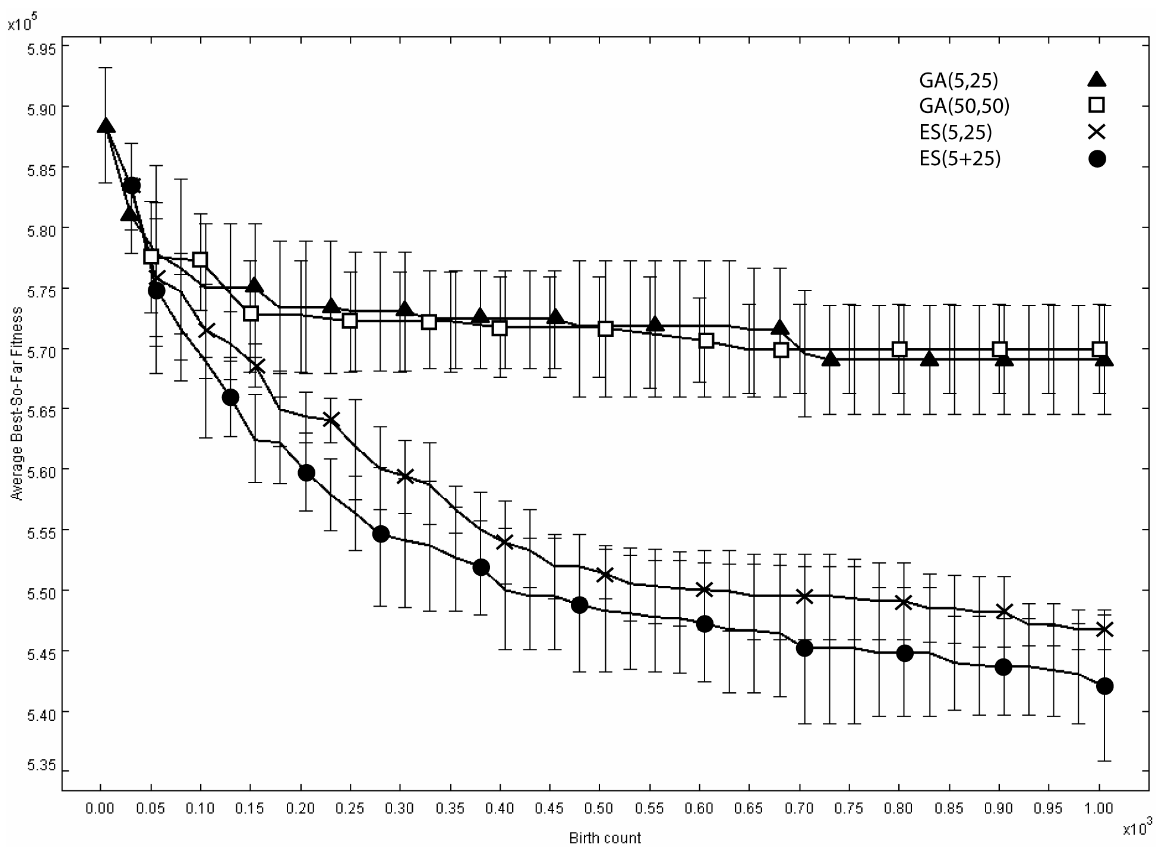


Figure 76. Comparison of the performance of GAs and ES in the optimization of a wind bracing system

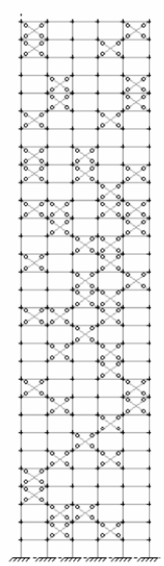
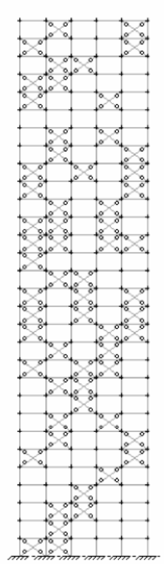
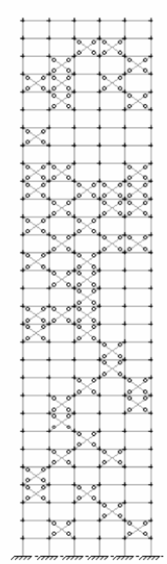
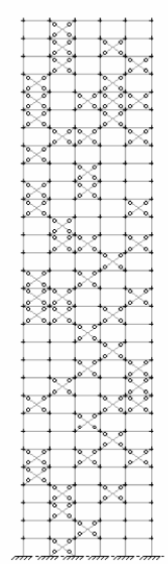
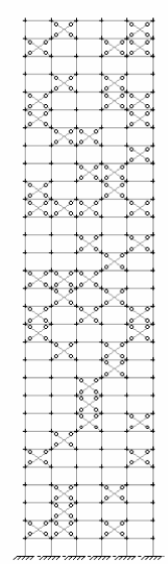
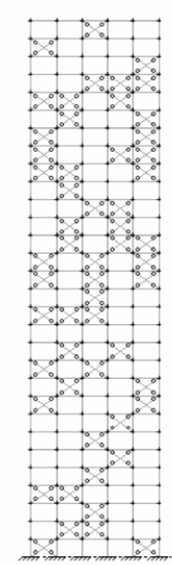
Concluding, the results of the design experiments revealed that ES performed better than GAs in this problem domain. Hence, they were employed in the design experiments reported in the remainder of this dissertation.

Optimal Designs

Optimal design concepts produced by the short-term evolutionary optimization processes with 2 types of wind bracing elements are presented separately for each group of wind bracing elements. First, results of the experiments are reported in which the group No.1 was used, i.e. no bracings and simple X bracings. Subsequently, I discuss the best design concepts obtained with the second group of wind bracing elements, i.e. no bracings and K bracings.

Short-term experiments with the group No.1 produced design concepts which outperformed not only the best design concepts produced by elementary CA rules (see Table 18 and Table 31) but also the design concepts known from the structural engineering literature (see Table 10). Table 62 shows 6 best design concepts of wind bracings systems consisting of simple X bracings and produced in the short-term experiments. The fitness of the best design found in short-term experiments was equal to 531,790 lbs. and was better by more than 18,500 lbs., or 3.5 percent, than the best design generated by elementary CA rules (see Table 31). It also outperformed the design concepts known from the structural engineering literature and shown in Table 10 by more than 15,500 lbs., or 2.8 percent.

Table 62. Best design concepts of wind bracing systems produced in the short-term evolutionary optimization experiments with two types of bracings (no bracings and simple X bracings)

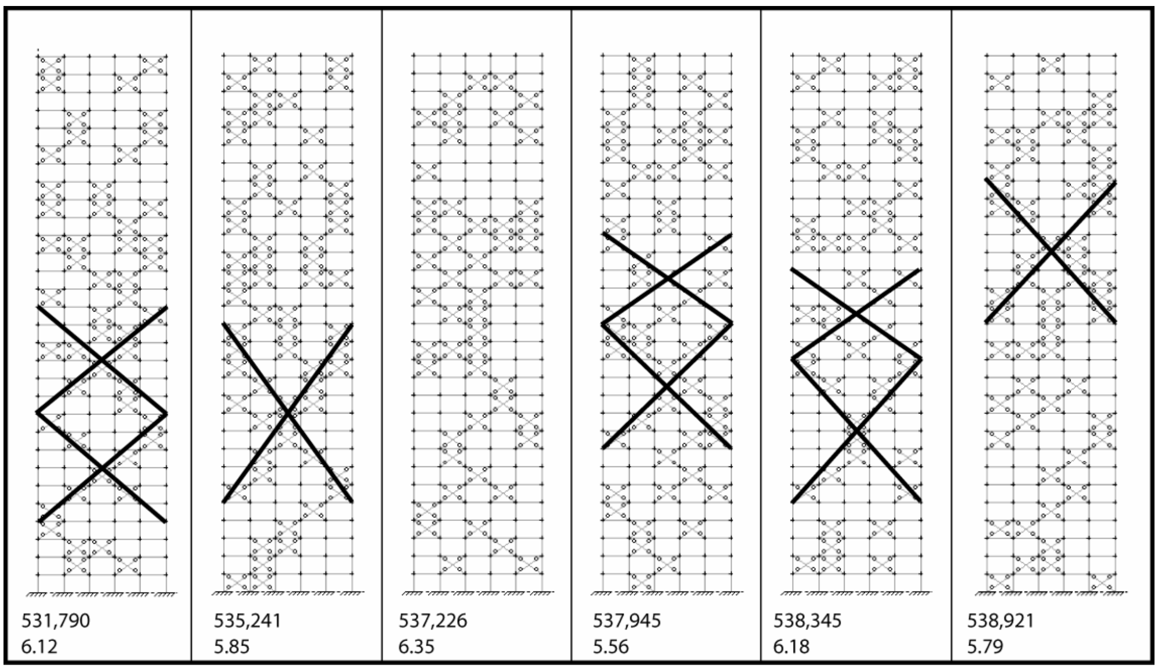
					
531,790 6.12	535,241 5.85	537,226 6.35	537,945 5.56	538,345 6.18	538,921 5.79

All design concepts shown in Table 62 satisfy provisions of the design codes regarding the maximum allowed horizontal displacement. Most rigorous provisions restrict the maximum horizontal displacement to $\frac{1}{600}$ of the height of the tall building. In these experiments, 30 story buildings were considered and the story height was equal to 14 feet. Thus, the maximum allowed horizontal displacement was equal to 8.4 inches. All design concepts presented in Table 62 have the maximum horizontal displacements smaller than this value and thus, satisfy the provisions of the design codes.

Structural Shaping Patterns

Table 63 shows another interesting phenomenon. Several design concepts shown in the table exhibit an emergent pattern of crossed macro bracings in the lower and/or middle part of the structural system. This pattern is similar to the one known from the structural engineering literature (see Design 5 in Table 10). Table 63 identifies several emergent patterns of crossed macro bracings shared by the best design concepts. They occur in 5 out of 6 design concepts shown in this table.

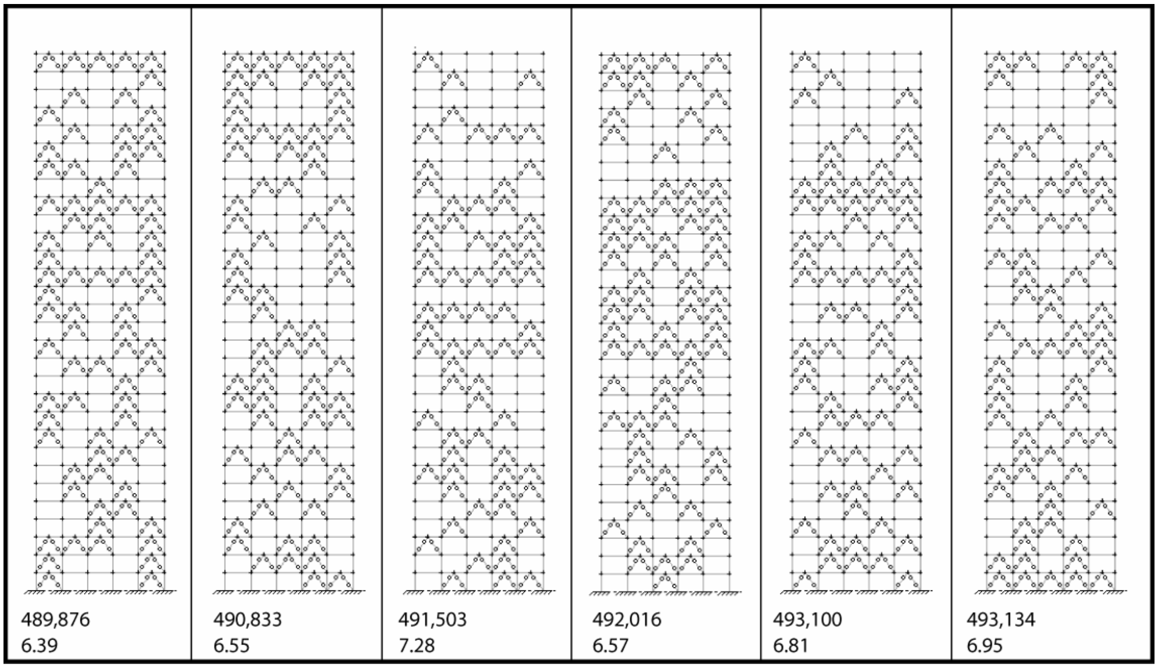
Table 63. Emergent patterns of crossed macro bracings in the best design concepts found in the short-term experiments



The experimental results obtained with the group No.2 were quite different. The best design concepts produced by evolutionary optimization processes were worse than the ones generated by elementary CA rules. Table 64 shows the best design concepts found in the short-term experiments. The fitness of the best design was equal to 489,876 lbs. and was about 40,000 lbs., or 9 percent, worse than the best design found in the experiments with elementary CAs (see Table 16 and Table 33). It is also difficult to identify any emergent pattern shared by several design concepts shown in Table 64.

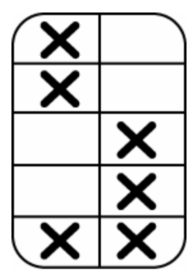
When we compare the best design concepts consisting of simple X bracings with the ones composed of K bracings, we observe that the latter produce design concepts of a significantly reduced total weight. On the other hand, they also exhibit larger horizontal displacements than the design concepts consisting of simple X bracings. These results are consistent with the previous findings reported in section 6.2.1.

Table 64. Best design concepts of wind bracing systems produced in the short-term evolutionary optimization experiments with two types of bracings (no bracings and K bracings)



In the next subsection, I will investigate the impact of the length of evolutionary optimization processes on the performance of the produced design concepts. Specifically, I will be interested in determining whether evolutionary optimization processes can find better design concepts of wind bracing systems composed of simple X bracings and what performance gain, if any, they can achieve. I will also try to determine whether longer evolutionary optimization processes can produce better design concepts of wind bracings systems consisting of K bracings than the ones produced by elementary CAs.

Long-term Evolutionary Optimization



In this group of experiments, long-term evolutionary optimization processes involving two types of wind bracing elements were conducted. As before, two groups of wind bracing elements were considered. The length of the long-term evolutionary optimization processes was significantly larger than the short-term processes and involved 10,000 fitness evaluations.

Extensive sensitivity analyses conducted during the short-term optimization processes were aimed to identify the best combination of evolutionary computation parameters for this problem domain. The results of these analyses were reported in the previous subsection. Based on that, the best combination of the evolutionary computation of parameters and their values was selected. It is presented in Table 65.

The same experimental parameters were used for both groups of wind bracing elements. The results of the long-term evolutionary optimization experiments are discussed below separately for each group of wind bracing elements.

Table 65. Evolutionary computation parameters and their values used in the long-term evolutionary optimization with two types of wind bracing elements

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies
Population sizes (parent, offspring)	(5,25)
Generational model	Overlapping ($\mu+\lambda$)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.025
Crossover (type, rate)	(uniform, 0.2)
Fitness	Total weight of the steel structure (determined by the first-order analysis)
Initialization method	Random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	10,000 fitness evaluations
Number of runs	5 in each experiment

Performance Improvement

Figure 77 shows the progress of the long-term evolutionary optimization process for the first group of wind bracing elements. It also compares the average best-so-far performance of the long-term optimization process to the average best-so-far fitness obtained after 1,000 evaluations (the short-term process). Figure 77 clearly shows that there is a significant performance improvement when evolutionary optimization processes are conducted for a larger number of fitness evaluations. However, sufficient computational resources must be available because the long-term processes take, in this case, 10 times longer than the short-term optimization processes[2].

The average performance improvement of the long-term processes with respect to the short-term processes was equal to about 15,500 lbs., or 3 percent. The improvement between the average fitness after 10,000 fitness evaluations and the average fitness of initial parents was equal to almost 62,000 lbs., or 10.5 percent.

Optimal Designs

Table 66 shows 6 best design concepts produced in the long-term experiments with the group No.1. They were produced in two independent runs (designs 1-4 in one run and designs 5-6 in the other run).

[2] Average computation time for the long-term experiments with 5 independent runs and 10,000 fitness evaluations was equal to 25 days on computers with Pentium4 2.0GHz processors and 512MB RAM.

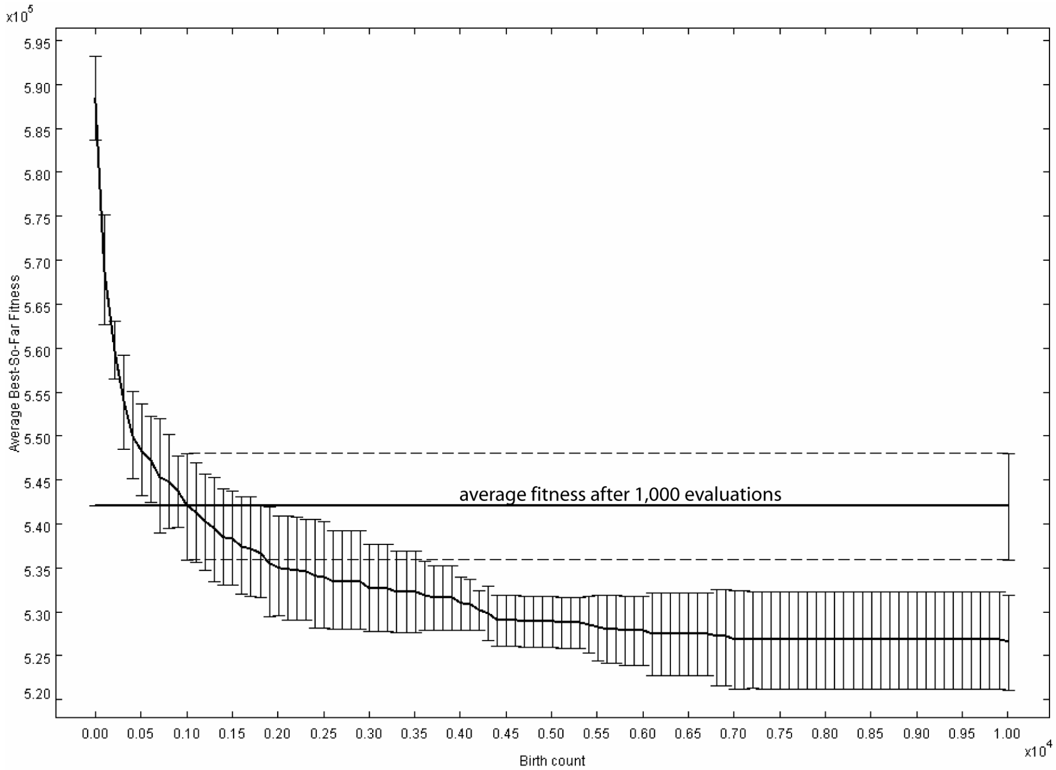
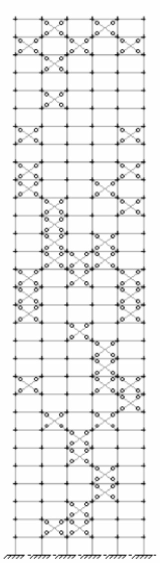
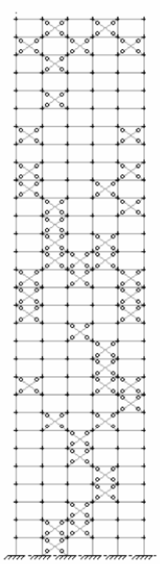
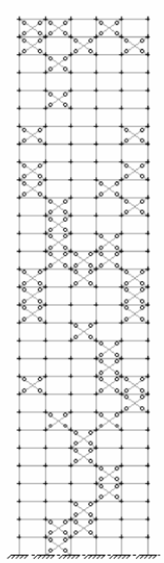
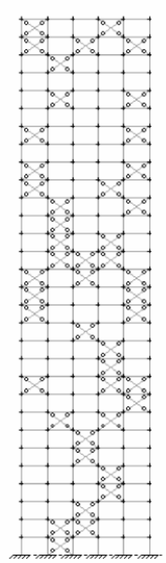
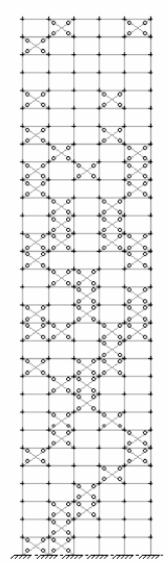
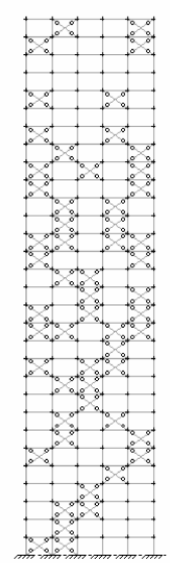


Figure 77. Average performance of the long-term evolutionary optimization processes with 2 types of wind bracing elements (no bracings and simple X bracings)

Table 66. Best design concepts of wind bracing systems produced in the long-term evolutionary optimization experiments with two types of bracings (no bracings and simple X bracings)

					
520,349 7.16	520,388 7.17	521,933 7.22	522,021 6.88	524,890 6.90	524,907 6.41

The fitness of the best design concept was equal to 520,349 lbs. and was more than 11,000 lbs., or 2 percent, better than the fitness of the best design produced in the short-term experiments (see Table 62). In general, all 6 best design concepts shown in Table 66 had better fitness than the best design concept produced in the short-term experiments.

Structural Shaping Patterns

Table 62 shows another interesting phenomenon, namely the emergence of two distinct patterns in different parts of the structural system. The previously identified crossed macro bracing pattern emerges in the lower part of the structure. A new pattern, which is similar to the pattern produced by rule 105 (see Table 35), emerges in the central part of the structural system.

Performance Improvement

Figure 78 shows the progress of the long-term evolutionary optimization in the experiments with the second group of wind bracing elements. Here, as before, the long-term processes produced better design concepts than the short-term processes. The average performance improvement between the long-term process and the short-term process was equal to 8,500 lbs., or 1.7 percent. It is only about half of the improvement which was achieved in the case of the first group of wind bracing elements.

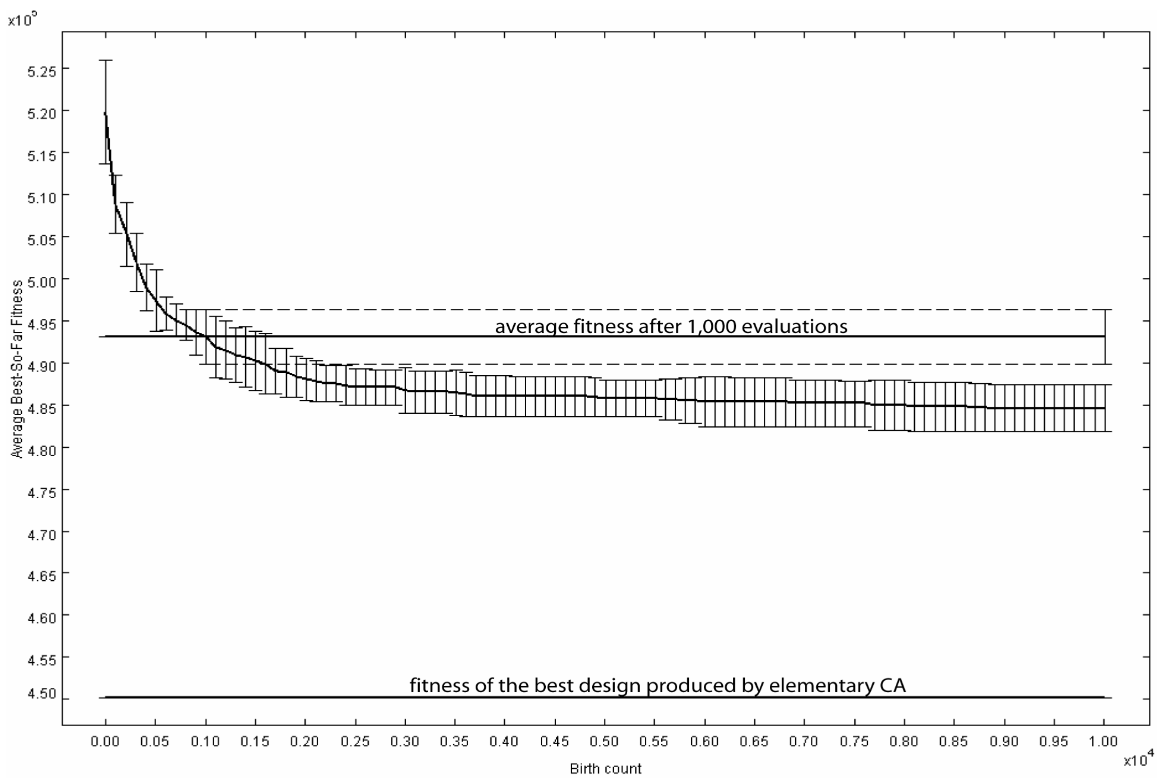
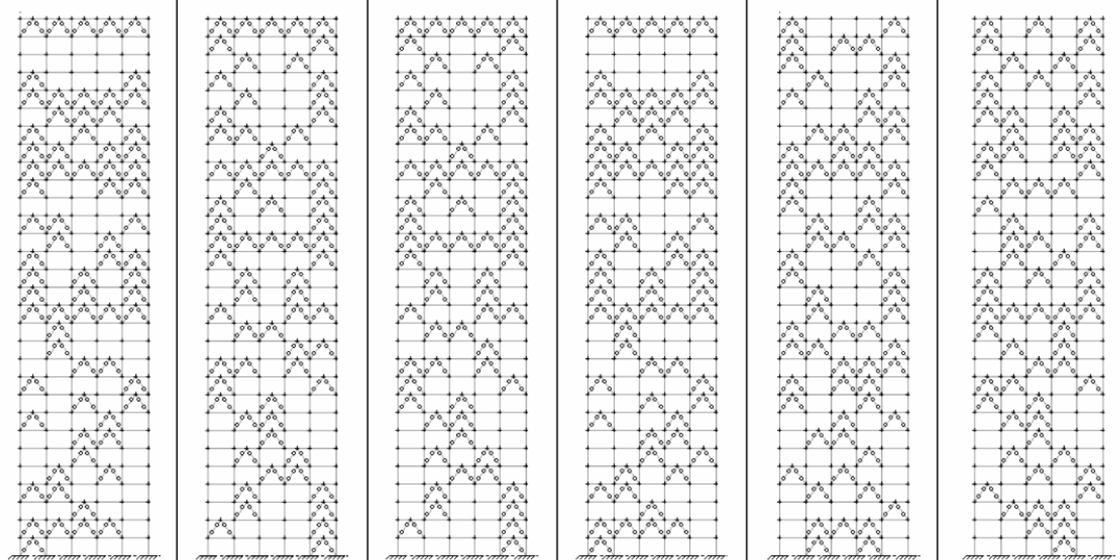


Figure 78. Average performance of the long-term evolutionary optimization processes with 2 types of wind bracing elements (no bracings and K bracings)

Optimal Designs

Figure 78 also shows that the long-term evolutionary optimization processes were not able to produce as good designs as the ones produced by elementary CAs. In fact, they were substantially inferior to the best design concept shown in Table 16. The best designs produced in the long-term experiments are presented in Table 67.

Table 67. Best design concepts of wind bracing systems produced in the long-term evolutionary optimization experiments with two types of bracings (no bracings and K bracings)



482,276 7.28	482,442 6.68	482,462 6.67	482,917 6.81	485,077 6.90	485,760 6.46

The fitness of the best design concept was equal to 482,276 lbs. It was more than 32,000 lbs. worse than the best design concept developed by an elementary CA. It is also difficult to identify any patterns which shared by several design concepts shown in Table 67.

Summary

In this section, I described the results of the experiments in which systems of wind bracings in tall buildings were optimized using evolutionary algorithms. In order to make fair comparisons of the results produced by evolutionary optimization processes and the ones produced by elementary CAs (see section 6.2), the number of types of wind bracing elements considered during the optimization processes was restricted to 2 in each experiment. Similarly as in section 6.2, two groups of wind bracing elements, each consisting of two types of wind bracings, were separately investigated in the evolutionary optimization experiments.

Besides, I also investigated the influence of several evolutionary computation parameters on the performance of the optimization processes, including the rates of mutation and crossover operators, sizes of parent and offspring populations, the type of the generational model (overlapping vs. nonoverlapping), and the type of the evolutionary algorithm (ES vs. GA). An extensive sensitivity analysis was conducted during the short-term processes to identify the optimal values of these parameters. They were later applied in the long-term evolutionary optimization processes.

The sensitivity analysis showed that the rate of mutation operator had a strong impact on the performance of the optimization processes when ES were employed. The lowest rate of the mutation operator, equal to 0.025, produced the best results. On the other hand, the impact of the crossover rate was negligible. Also, the sizes of parent and offspring populations significantly affected the performance of ES but had limited impact on the performance of GAs in this problem domain. Generally, ES with small sizes of parent of offspring populations significantly outperformed the ones with large populations.

The impact of the type of the generational model was studied in the context of two kinds of ES: ES(5+25) (overlapping) and ES(5,25) (nonoverlapping). The results of these experiments revealed that both algorithms produce comparable results. Finally, the performance of two evolutionary algorithms was compared, namely ES and GAs. The experimental results showed that ES outperformed GAs by a large margin in this problem domain.

Optimal design concepts of wind bracing systems produced in the short-term experiments were also presented and discussed. The average performance improvement (difference in average performance of the design concepts at the end of evolutionary optimization processes and of the initial parents) achieved in the short-term optimization processes was equal to 46,461 lbs., or 7.9 percent, in the case of the first group of wind bracing elements (simple X bracings). The results produced with the group No.2 (K bracings) were slightly worse and equal to 26,633 lbs., or 5.1 percent. Later, in the final subsection, I reported that the average performance improvement increased in the long-term processes and exceeded 61,000 lbs., or 10 percent, in the case of simple X bracings, and 35,000 lbs., or 6 percent, for K bracings. These results are illustrated in Figure 79.

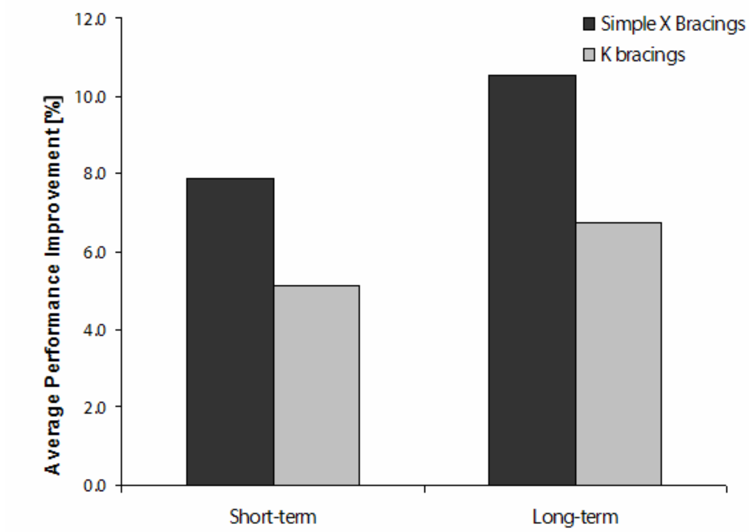


Figure 79. Comparison of the average performance improvements produced in the evolutionary optimization of wind bracing systems with 2 types of bracing elements in the short-term and long-term experiments

During the performance analysis phase, I also compared the best design concepts produced in the evolutionary optimization processes with the ones generated using generative representations based on elementary CAs (see section 6.2). The obtained results differed for each group of wind bracing elements. Evolutionary optimization processes produced better results than generative representations for simple X bracings. However, the opposite results were achieved for K bracings. Here, the best design concept generated by elementary CAs significantly outperformed (by more than 8 percent) the best design concept found in the evolutionary optimization experiments. As reported in the final subsection, even the long-term processes did not produce better design concepts composed of K bracings than the ones generated by elementary CA rules. These comparisons are presented graphically in Figure 80.

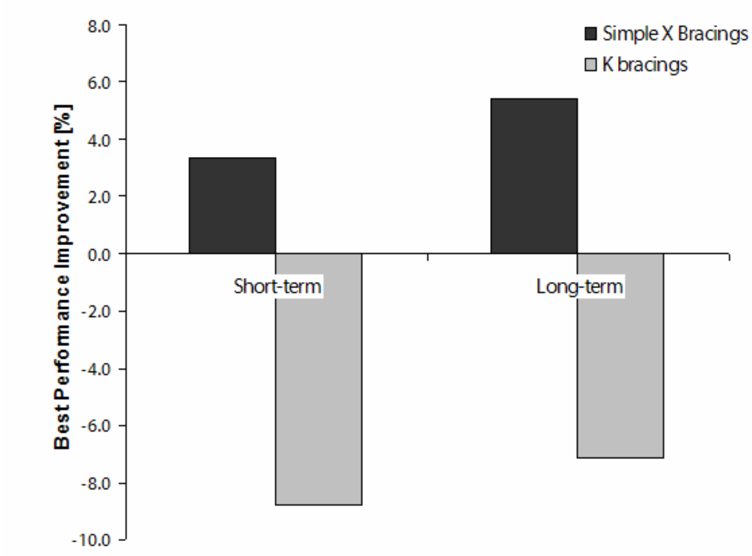
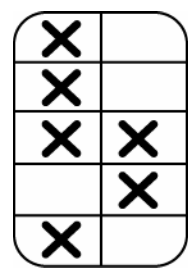


Figure 80. Comparison of the performance improvements between the best designs produced in the evolutionary optimization experiments and the best designs generated by elementary CAs (negative values correspond to situations in which elementary CAs produced better design concepts than evolutionary optimization processes)

In the next section, I will empirically test whether better design concepts of wind bracing systems can be produced when the entire selection of 7 types of wind bracing elements is used during the optimization process.

7.2.2. Optimization with Seven Types of Wind Bracings



In this section, I describe results of the design experiments in which 7 types of wind bracing elements were used (see Figure 19) in the evolutionary optimization processes. As it was the case with the experiments reported in the previous section, a sensitivity analysis was conducted first during the short-term processes in order to determine the most suitable combination of evolutionary computation parameters. Once this combination of parameters has been found, it was used in the long-term optimization experiments.

Short-term Evolutionary Optimization

Table 68 shows the evolutionary computation parameters a used in the short-term experiments. Only one type of evolutionary algorithm was used in the design experiments with 7 types of wind bracing elements. Experimental results reported in the previous section indicated that GAs produce inferior results to ES in this problem domain. Thus, ES were selected as the only evolutionary algorithm which conducted evolutionary optimization processes. Two kinds of ES were used: ES(μ+λ) and ES(μ,λ).

The results reported in the previous section have shown that large population sizes produce inferior results in this problem domain. Hence, only small population sizes were used in the design experiments with 7 types of wind bracing elements. Two combinations of parent and offspring population sizes, i.e. (1,5) and (5,25), were investigated for ES(μ+λ) and one combination, i.e. (5,25), for ES(μ,λ).

Six combinations of mutation and crossover rates were studied experimentally in the short-term experiments. High mutation and crossover rates, i.e. mutation and crossover rates equal to 0.5, were excluded from the set of rates investigated in this section because they previously produced inferior results.

Table 68. Evolutionary computation parameters and their values used in the short-term optimization experiments with 7 types of wind bracing elements

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies (ES)
Generational model	Overlapping for ES(μ+λ), Nonoverlapping for ES(μ,λ)
Population sizes (parent, offspring)	(1,5), or (5,25) for ES(μ+λ) (5,25) for ES(μ,λ)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.025, 0.1, or 0.3
Crossover (type, rate)	(uniform, 0), (uniform, 0.2)
Fitness	Total weight of the steel structure (determined by the first-order analysis)
Initialization method	Random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	1,000 fitness evaluations
Number of runs	5 in each experiment

The same fitness measure was used, i.e. the total weight of the steel structural system calculated using the first-order structural analysis. As before, infeasible designs were assigned the fitness value of 0. Each design experiment involving a single combination of parameter values consisted of 5 independent runs. Each run was terminated after 1,000 fitness evaluations.

Optimal Mutation and Crossover Rates

Figure 81 shows typical results regarding the impact of the mutation rate on the performance of the evolutionary algorithm. In this case, experimental results obtained using ES(5,25) are presented with the crossover rate equal to 0.2. As before, the lowest mutation rates produced the best results. Figure 81 clearly shows that the best optimization progress was achieved when the mutation rate was equal to 0.025 and it decreased when higher mutation rates were applied. No such pattern was observed for crossover rates. On the contrary, similar results were obtained with various crossover rates. These findings are analogical to the ones reported in the previous section for evolutionary optimization processes with 2 types of wind bracing elements.

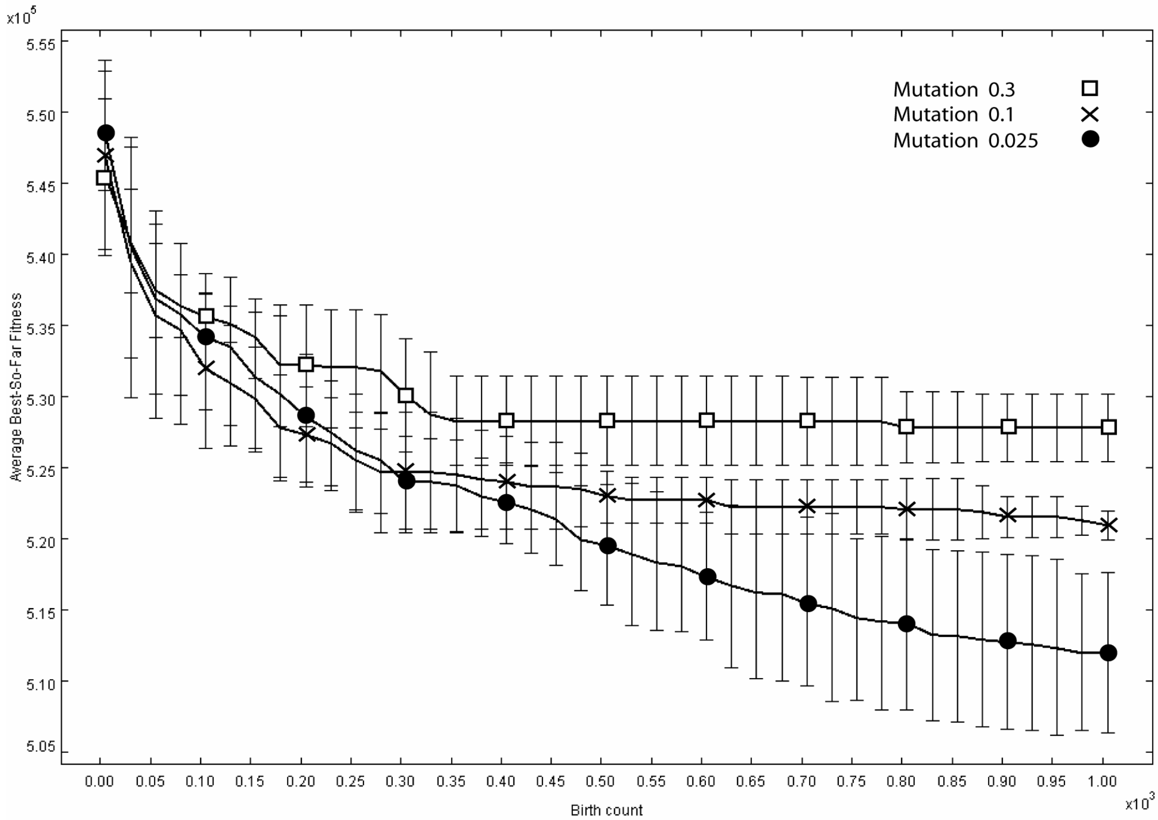


Figure 81. The influence of the rate of mutation operator on the progress of the short-term evolutionary optimization processes when 7 types of wind bracing elements were used

Performance Improvement

Figure 82 compares the performance of the short-term evolutionary optimization with 7 types of wind bracing elements to the results achieved when 2 types of bracing elements were used. Here, ES(5+25) was used with the mutation and crossover rates equal to 0.025 and 0.2, respectively. It is clear that when 7 types of wind bracing elements are used, the fitness of produced design concepts is, on average, better than the design concepts consisting of simple X bracings and worse than the design concepts composed of K bracings.

The average fitness obtained after 1,000 evaluations in the experiments with 7 types of bracing elements was equal to 511,480 lbs. and was more than 30,500 lbs. better than the average fitness obtained in the experiments with simple X bracings. At the same time, it was more than 18,500 lbs. worse than the value obtained in the experiments with K bracings.

The overall performance improvements in the short-term experiments shown in Figure 82 for 7 types of wind bracings, simple X bracings, and K bracings were equal to 6.1%, 7.9%, and 5.1% percent, respectively.

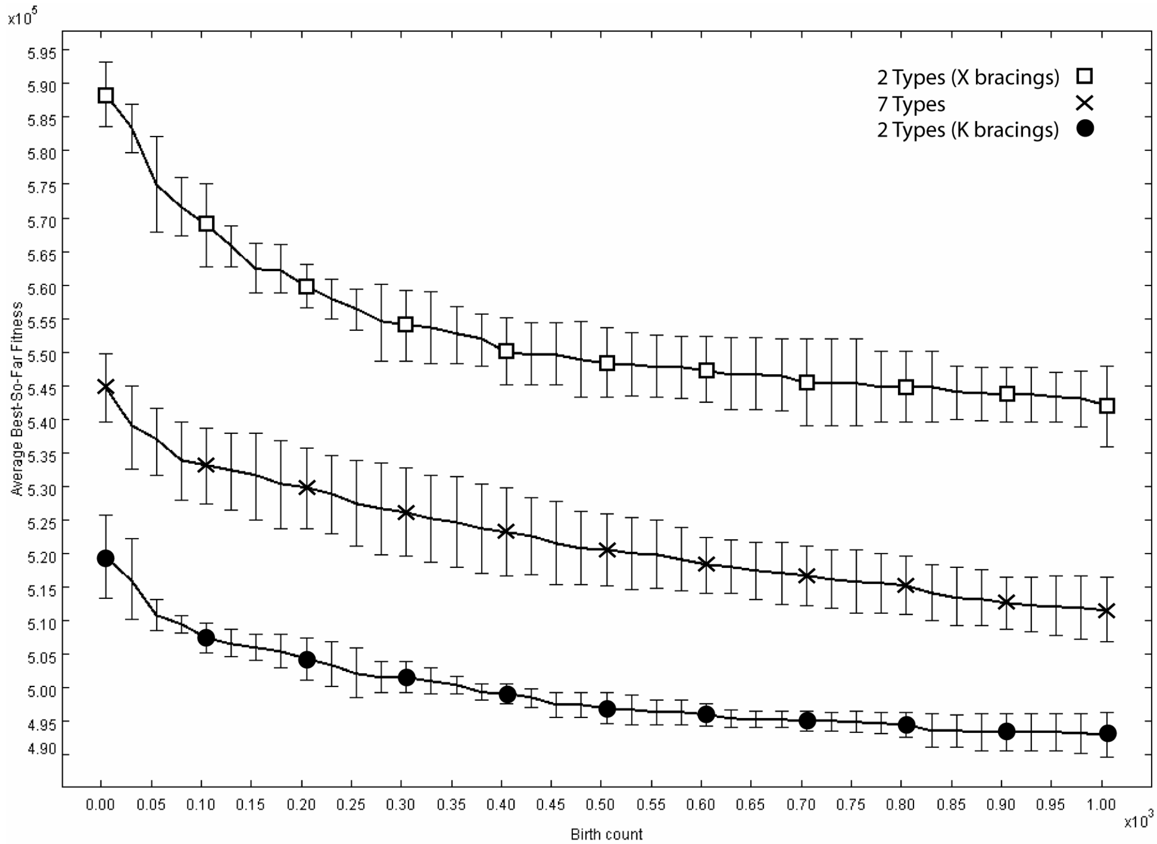


Figure 82. Comparison of the performance of the evolutionary optimization when 2 types and 7 types of wind bracing elements were used

Optimal Designs

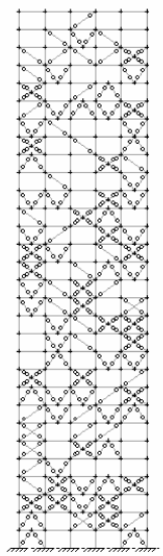
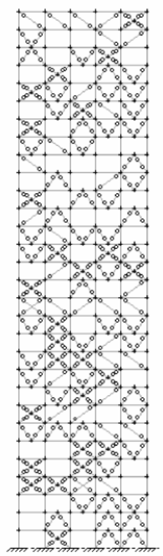
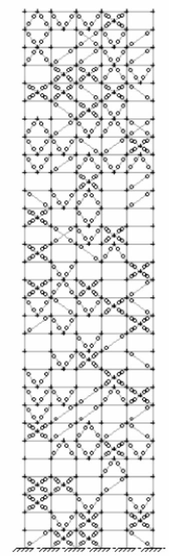
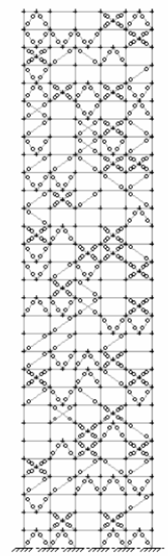
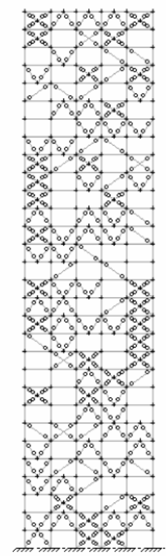
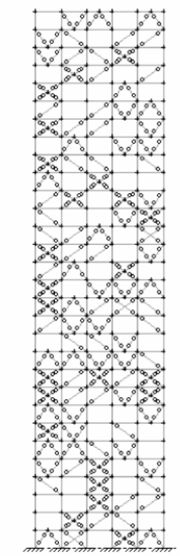
These findings were further confirmed by the results presented in Table 69. It shows the best design concepts produced in the short-term experiments with 7 types of wind bracing elements. The fitness of the best design concept found in the short-term experiments was equal to 504,162 lbs. and was more than 27,500 lbs better than the fitness of the best design concept produced in the experiments with simple X bracings (see Table 62). On the other hand, it was more than 14,000 lbs. worse than the best design concept composed of K bracings (see Table 64).

All design concepts shown in Table 69 exhibit rather random-looking configurations of wind bracing elements. It is hard to identify any structural shaping patterns shared by the design concepts.

Long-term Evolutionary Optimization

The sensitivity analysis conducted in the short-term experiments helped identify the best combination of evolutionary computation parameters and their values. It revealed that the best combination included exactly the same parameters and values as the ones used in the long-term processes with 2 types of wind bracing elements. Thus, the parameters shown in Table 65 were also used in the long-term experiments with 7 types of wind bracing elements.

Table 69. Best design concepts of wind bracing systems produced in the short-term evolutionary optimization experiments with 7 types of bracing elements

					
504,162 5.69	505,137 5.65	505,230 5.83	506,115 5.62	507,538 5.56	508,293 5.65

Performance Improvement

Figure 83 shows the progress of the long-term evolutionary optimization in the experiments with 7 types of wind bracing elements and compares it to the average fitness obtained after 1,000 evaluations. The average performance improvement between the long-term processes and short-term processes was equal to about 21,900 lbs., or 4.3 percent, and it was the largest performance improvement in the long-term experiments reported so far. The difference between the average fitness after 10,000 fitness evaluations and the average fitness of the initial parents was equal to more than 55,500 lbs., or 10.2 percent.

There are two major qualitative differences between the average best-so-far curve shown in Figure 83 and the curves obtained in the long-term experiments with 2 types of wind bracing elements (see Figure 77 and Figure 78). First, Figure 83 shows that there is a sustained evolutionary optimization progress during the entire run when 7 types of wind bracing elements are used. On the contrary, the best-so-far curves in the optimization experiments with 2 types of wind bracing elements level off much faster during the long-term runs. Second, the experimental results obtained with 7 types of wind bracing elements show much larger variance than the ones with 2 types of wind bracing elements. This suggests that there is much larger amount of exploration of the design space being performed even in the late stages of the optimization process.

These differences can be easily identified in Figure 84 which compares the 3 long-term experiments. It clearly shows that the average best-so-far curves for the experiments involving 2 types of wind bracing elements level off after about 7,000 fitness evaluations. It is not the case with the curve representing the results of the experiment with 7 types of wind bracing elements which shows steady optimization progress throughout the entire run.

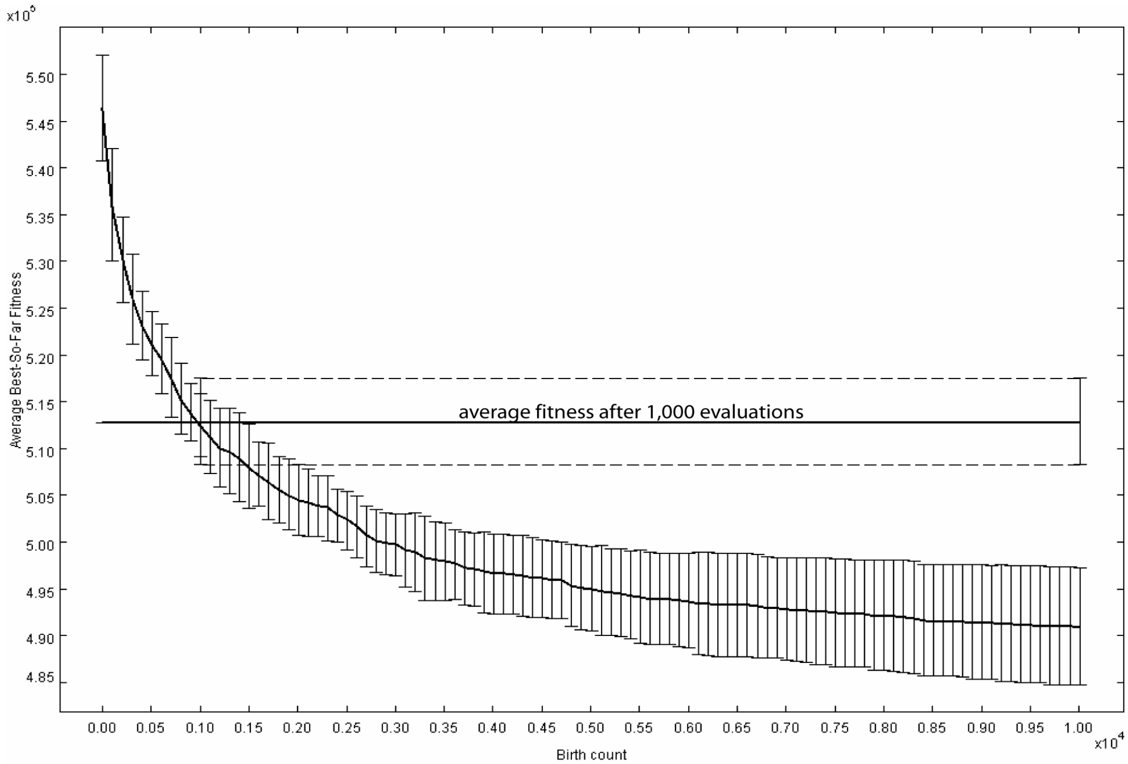


Figure 83. Average best-so-far performance of the long-term evolutionary optimization processes with 7 types of wind bracing elements

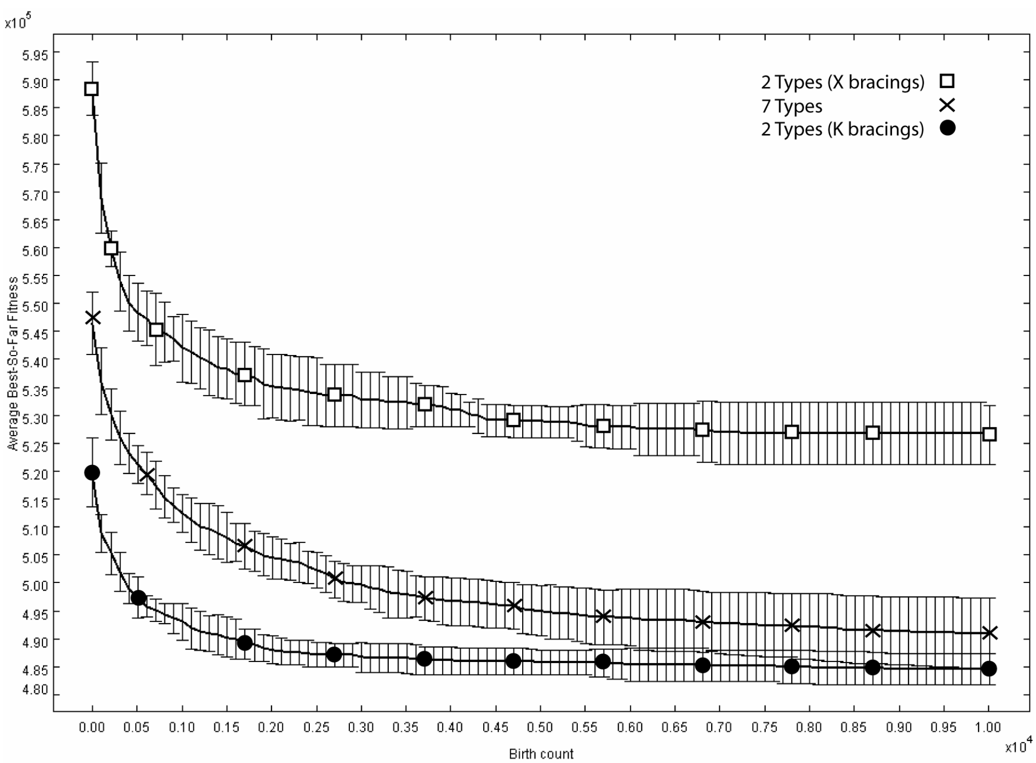
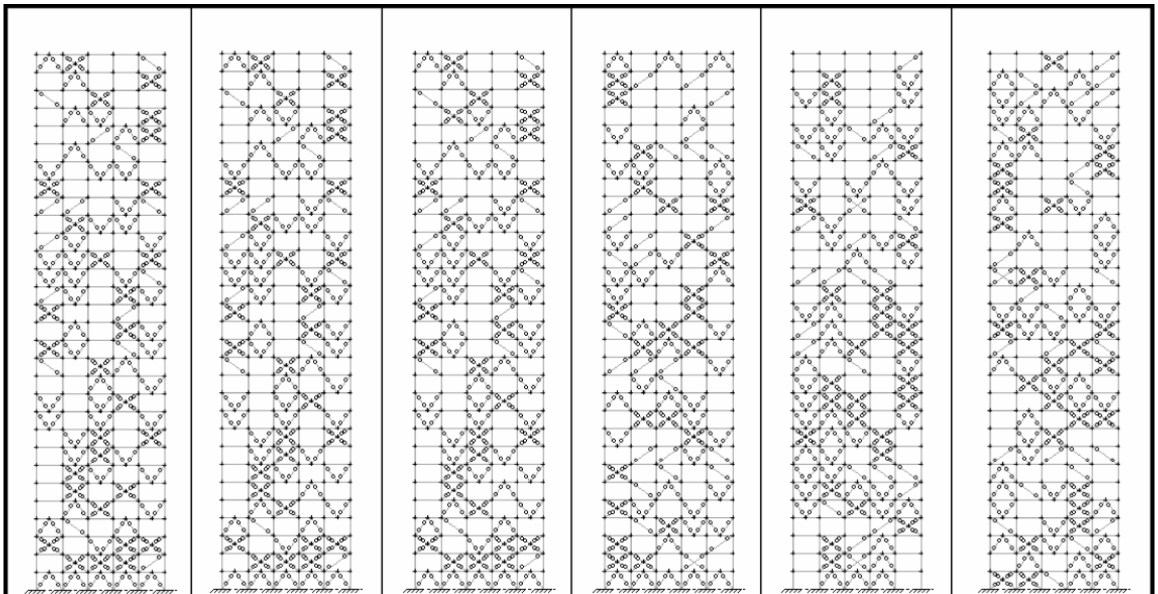


Figure 84. Comparison of the evolutionary optimization progress in the long-tem experiments with 2 and 7 types of wind bracing elements

Optimal Designs

Table 70 presents 6 best designs produced in the long-term experiments with 7 types of wind bracings. The fitness of the best design was equal to 485,081 lbs. and was more than 19,000 lbs. better than the fitness of the best design produced in the short-term experiments. It was only slightly worse (about 3,000 lbs.) than the fitness of the best design concept produced in the long-term experiments with K bracings.

Table 70. Best design concepts of wind bracing systems produced in the long-term evolutionary optimization experiments with 7 types of bracing elements



485,081 6.14	485,175 6.14	485,293 6.14	489,137 6.21	492,453 5.98	498,520 6.03

Summary

In this section, I reported the results of the design experiments in which the entire selection of 7 types of wind bracing elements was used to optimize the topology of a wind bracing system in a tall building. Similarly as in the previous section, both short-term and long-term design experiments were performed.

The sensitivity analysis conducted during the short-term processes revealed that the same evolutionary computation parameters which worked well in the design experiments with 2 types of wind bracing elements (see section 7.2.1) also produced the best results in the experiments with 7 types of wind bracing elements.

The short-term experiments showed that the best design concepts produced by evolutionary optimization processes with 7 types of wind bracing elements are better (in terms of the total weight) than the ones consisting of simple X bracings but worse than the design concepts composed of K bracings.

Figure 85 compares the average performance improvements achieved in the evolutionary optimization experiments with 7 and 2 types of wind bracings elements. It shows that the improvement of 6.4 percent obtained in the short term-experiments with 7 types of wind bracing elements locates them in the middle of a performance improvement range achieved for simple X bracings (7.9 percent) and K bracings (5.1 percent). The performance significantly increases in the long-term experiments (the improvement exceeds 10 percent) and almost reaches the level obtained in the long-term experiments with simple X bracings (10.5 percent).

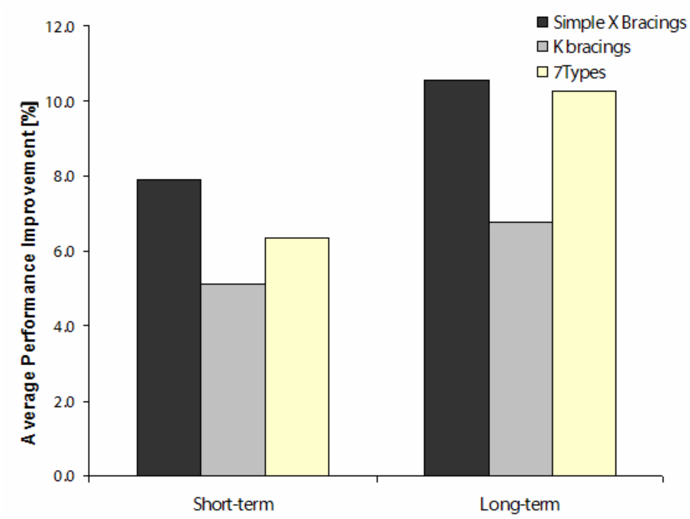


Figure 85. Comparison of the average performance improvements produced in the evolutionary optimization of wind bracing systems with 7 and 2 types of bracing elements in the short-term and long-term experiments

Figure 86 compares the performance improvements between the best design concepts produced in the evolutionary optimization processes (with 7 and 2 types of wind bracing elements) with the best ones generated using generative representations based on elementary and one-dimensional CAs (sections 6.2 and 6.3). It shows that also in the case of the optimization with 7 types of bracing elements, evolutionary algorithms produced inferior design concepts as the ones generated by one-dimensional CAs. The differences exceeded 12 and 8 percent in the short-term and the long-term experiments, respectively.

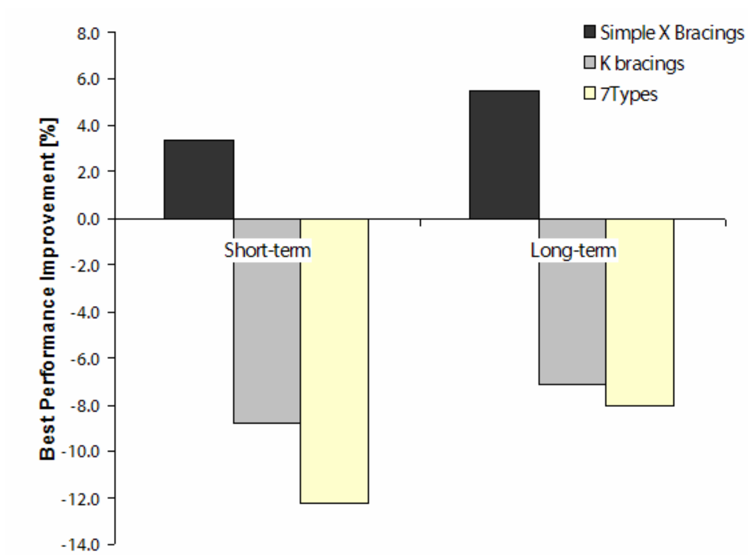
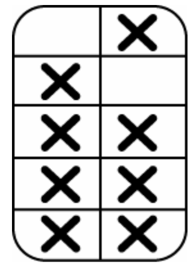


Figure 86. Comparison of the performance improvements between the best designs produced in the evolutionary optimization experiments and the best designs generated by elementary and one-dimensional CAs

In the next section, I will experimentally investigate the evolutionary computation component of EED in optimizing the entire steel structural systems in tall buildings. In these experiments, not only the topology of the wind bracing system was optimized but also optimal configurations of beams and supports were sought.

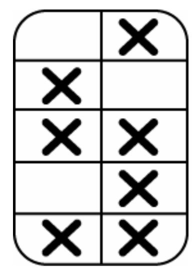
7.3. Optimization of the Entire Steel Structural Systems



In this section, I will empirically investigate the usefulness of the evolutionary computation component of EED in optimizing more complex engineering systems, namely the entire steel structural systems in tall buildings.

The experimental work reported in this section has been divided into two parts to investigate whether or not domain knowledge encoded in the initial parents improves the performance of the evolutionary optimization processes. First, in subsection 7.3.1, I will investigate evolutionary optimization processes which were initialized with randomly generated parents. On the other hand, in subsection 7.3.2, I will experimentally study evolutionary optimization processes in which background knowledge of the problem domain was used to initialize evolutionary algorithms. Specifically, a set of designs known from the structural engineering literature was be used as the initial population of parents. The results produced in both groups of experiments will be subsequently compared.

7.3.1. Starting from Randomly Generated Designs

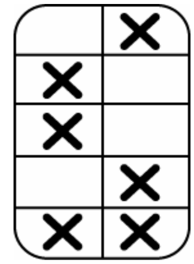


Experiments reported in this subsection considered the optimization of the entire steel structural systems in tall buildings. The evolutionary optimization processes were initialized, as before, with randomly generated parents. Thus, no domain knowledge was added in this section to start evolutionary optimization processes.

In the conducted experiments, 7 types of wind bracing elements (see Figure 19), two types of beams (see Figure 20), and two types of supports (see Figure 22) were considered. Columns, however, were kept the same during the entire evolutionary optimization processes. Table 71 shows parameters of the design problem studied in this subsection.

As in the previous sections, 30 story buildings with 5 bays were considered. Also, two groups of experiments were performed: short-term and long-term. Results of both groups of experiments are reported in the following subsections.

Short-term Evolutionary Optimization



In this group of experiments, the short-term evolutionary optimization of the entire steel structural systems in tall buildings was conducted. Evolutionary computation parameters used in these experiments are presented in Table 72.

Table 72 shows that ES with the overlapping generational model, i.e. ES(μ+λ), was employed in the experiments reported in this subsection. Furthermore, three combinations of parent and offspring population sizes were investigated. For each combination of population sizes, an extensive parameter search was conducted involving 9 combinations of mutation and crossover rates.

Table 71. Problem parameters and their values used in the conducted experiments

Problem Parameter	*Value(s)*
Problem type	Design of the entire steel structural system in a tall building
Number of stories	30
Number of bays	5
Bay width	20 feet (6.01 m)
Story height	14 feet (4.27 m)
Distance between transverse systems	20 feet (6.01 m)
Types of bracing elements	No, Diagonal \, Diagonal /, K, V, Simple X, and X
Types of beam elements	Pinned-Pinned, and Fixed-Fixed
Types of column elements	Fixed-Fixed (only)
Types of supports	Pinned, and Fixed

Table 72. Evolutionary computation parameters and their values used in the short-term optimization experiments of the entire structural systems in tall buildings

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies (ES)
Generational model	Overlapping for ES(μ+λ)
Population sizes (parent, offspring)	(1,5), (5,25), or (50,250)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.025, 0.1, or 0.3
Crossover (type, rate)	(uniform, 0), (uniform, 0.2), or (uniform, 0.5)
Fitness	Total weight of the steel structure (determined by the first-order analysis)
Initialization method	Random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	1,000 fitness evaluations
Number of runs	5 in each experiment

The design processes were repeated 5 times for each combination of parameter values using a different value of the random seed each time. Each run was terminated after 1,000 fitness evaluations. The initial population of parents was generated randomly in every experiment reported in this subsection. Each design concept was represented as a fixed-length genome with 306 genes. In this case, however, the genome was nonhomogeneous, i.e. it contained genes which encoded attributes that didn't have the same number of possible values. More specifically, 150 genes representing types of wind bracing elements had 7 possible values, 150 genes representing types of beams had 2 possible values, and 6 genes representing types of supports had 2 possible values.

As previously, the fitness of each design concept was determined by the total weight of the steel structural system calculated using the first-order structural analysis. Whenever an infeasible design concept was generated, it was assigned the fitness value of 0 (death penalty).

The following subsections describe the obtained results.

Optimal Evolutionary Computation Parameters

As before, one of the major goals of the short-term experiments was to determine the optimal values of the evolutionary computation parameters, i.e. the values which provided the best optimization progress. These values were subsequently used in the long-term optimization experiments.

The results of the short-term experiments revealed that the same parameters which proved to perform well in the evolutionary optimization of wind bracing systems also worked best in optimizing the entire steel structural systems in tall buildings. For example, Figure 87 shows the impact of the parent of offspring population sizes on the fitness of produced design concepts. The average best-so-far fitness curves shown in Figure 87 are very similar to the ones obtained in the short-term optimization of wind bracing systems (see Figure 73).

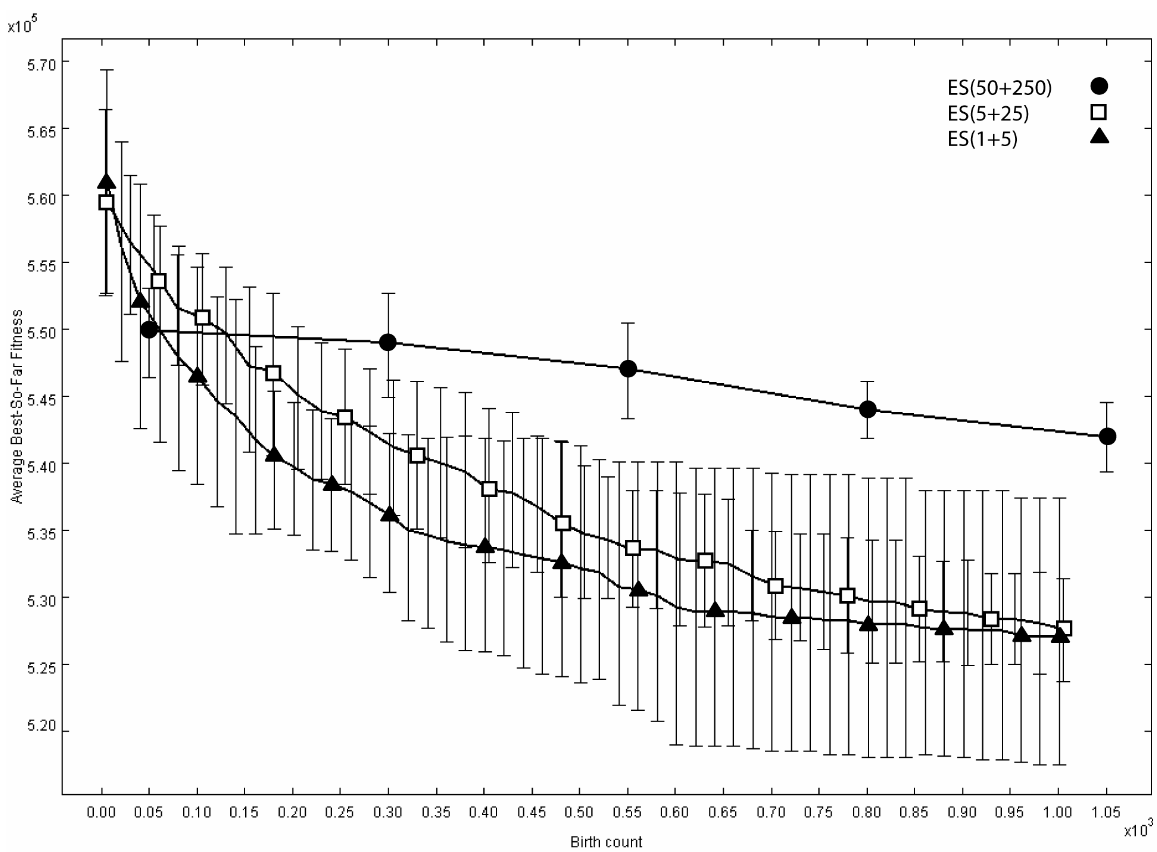


Figure 87. Influence of the sizes of parents and offspring populations on the progress of the short-term optimization of the entire steel structural systems in tall buildings

Figure 87 shows that ES with smaller population sizes significantly outperformed the ones with large population sizes in the short-term experiments. On the other hand, there are no significant differences between ES(5,25) and ES(1,5) in terms of the average best-so-far performance of the produced design concepts. The latter, however, introduced much larger variance than the former. These findings fully agree with the previous results obtained in the evolutionary optimization of wind bracing systems (see section 7.2).

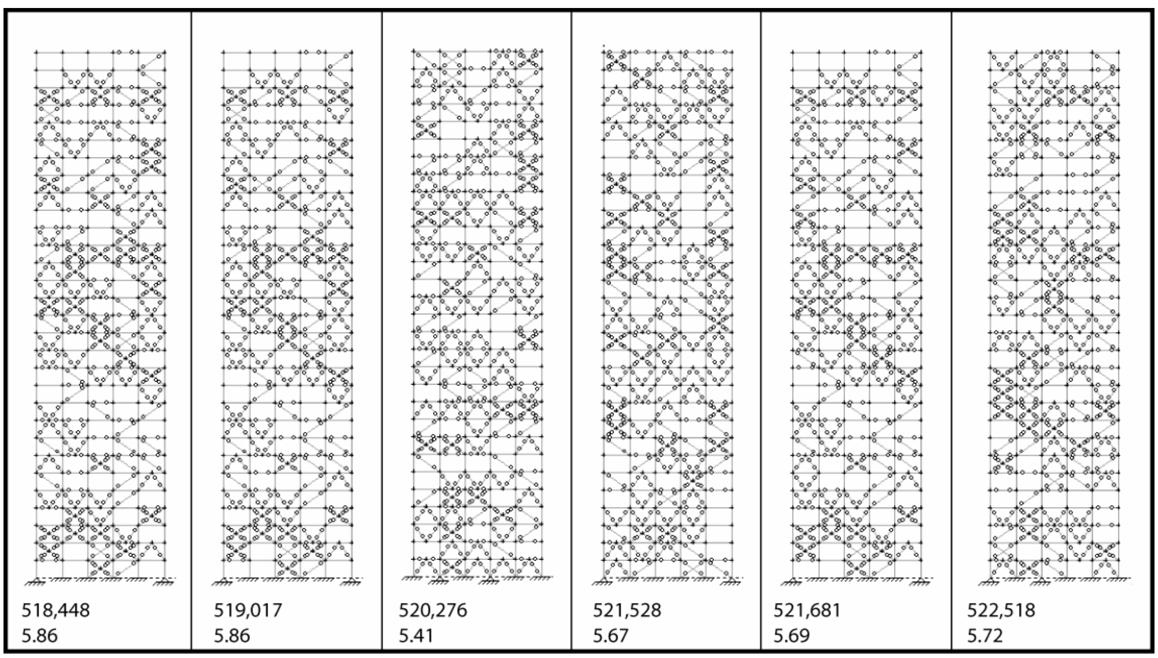
Similarly, the best evolutionary optimization progress was achieved when the lowest mutation rates were used. The experimental results showed similar relationships between the mutation rates and the average best-so-far fitness of the produced design concepts, as the ones presented earlier in Figure 67. The graph showing these findings was, however, omitted. Thus, the same parameter values as the ones included in Table 65 were selected for the long-term experiments.

Optimal Designs

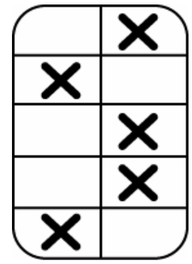
Table 73 shows the best design concepts of the entire steel structural systems in tall buildings produced in the short-term evolutionary optimization experiments. The fitness of the best design concept was equal to 518,448 lbs. and was more than 58,500 lbs., or 12 percent, worse than the best design produced by the generative representations based on multiple 1D CAs (see Table 57 in section 6.5.2).

Similarly as in Table 69, the best design concepts of the entire steel structural systems presented in Table 73 exhibit randomly looking configurations of structural members. In a few cases, however, in the middle part of the structural system, a macro bracing patterns formed by simple X bracings and X bracings begin to emerge.

Table 73. Best design concepts of the entire structural systems in tall buildings produced in the short-term evolutionary optimization experiments



Long-term Evolutionary Optimization



In this group of experiments, the long-term optimization of the entire steel structural systems in tall buildings was conducted using the evolutionary computation parameters presented in Table 65.

Performance Improvement

Figure 88 shows the average best-so-far performance of the long-term evolutionary optimization processes. It also compares the obtained results to the average outcomes produced in the short-term optimization experiments (after 1,000 evaluations) and the fitness of the best design concept generated by multiple 1D CAs (see Table 57).

The average fitness of the design concepts produced in the long-term experiments equaled 502,879 lbs. and was more than 21,500 lbs., or 4.1 percent, better than the average fitness obtained in the short-term experiments. The overall progress rate gained in the long-term experiments was, on average, equal to more than 58,000 lbs., or 10.3 percent, compared to about 36,000 lbs., or 6.4 percent, achieved in the short-term optimization experiments.

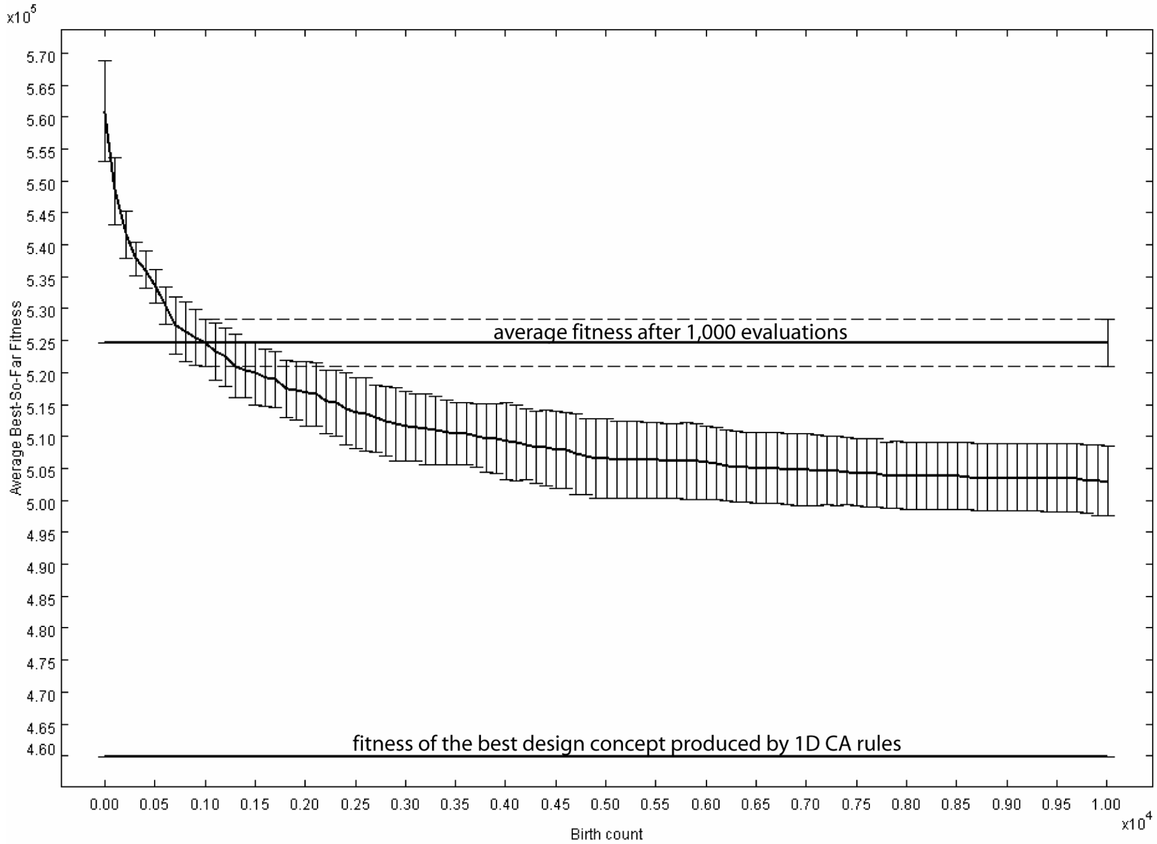


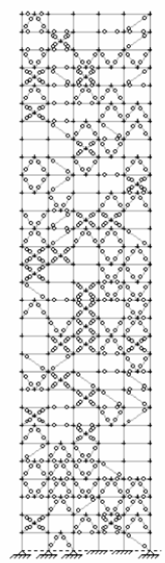
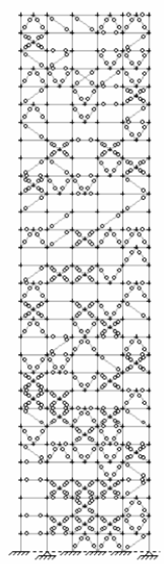
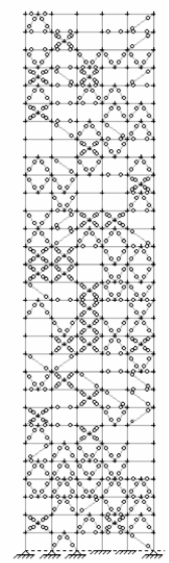
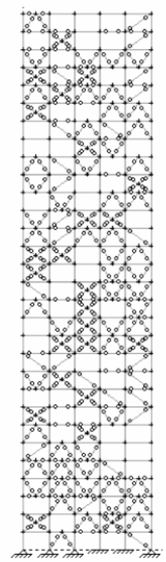
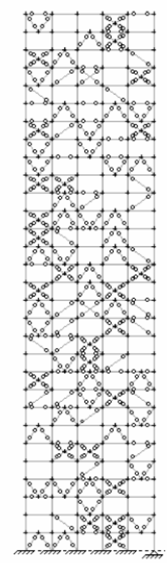
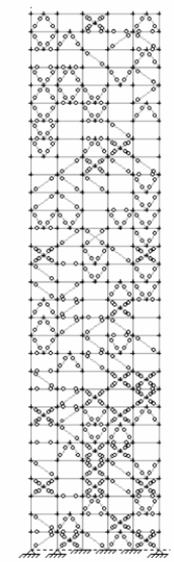
Figure 88. Average best-so-far fitness of the entire steel structural systems in tall buildings obtained in the long-term evolutionary optimization processes

On the other hand, the long-term evolutionary optimization processes did not produce design concepts of steel structural systems whose fitness was even close to the total weight found in the experiments with generative representations based on multiple 1D CAs (see section 6.5).

Optimal Designs

The best design concepts of the entire steel structural systems produced in the long-term evolutionary optimization experiments are presented in Table 74.

Table 74. Best design concepts of the entire structural systems in tall buildings produced in the long-term evolutionary optimization experiments

					
498,917 5.88	499,588 6.33	500,620 5.92	500,640 5.83	501,742 5.98	504,259 5.85

The fitness of the best design concept was equal to 498,917 lbs. and was about 39,000 lbs., or 8.5 percent, worse than the fitness of the best design concept generated by multiple 1D CA rules (see Table 57). Similarly as it was the case with design concepts produced in the short-term experiments, it is difficult to identify any structural shaping patterns in Table 74. All of them exhibit randomly looking configurations of wind bracings, beams, and supports.

Summary

In this section, I studied evolutionary optimization of the entire steel structural systems in tall buildings. As in the previous section, the optimization processes reported here were initialized randomly. Also, short-term and long-term design experiments were conducted. The short-term processes showed that the same evolutionary computation parameters identified as optimal in the previous sections can be successfully used to optimize the entire steel structural systems in tall buildings.

The average performance improvement achieved during the short-term experiments exceeded 6 percent while the corresponding improvement for the long-term experiments was greater than 10 percent. They were almost identical to the improvement levels achieved during the optimization of the wind bracing systems with 7 types of wind bracing elements (see section 7.2.2). Figure 89 compares the average performance improvements obtained in the experiments reported in this section with the ones reported earlier in the evolutionary optimization of wind bracing systems.

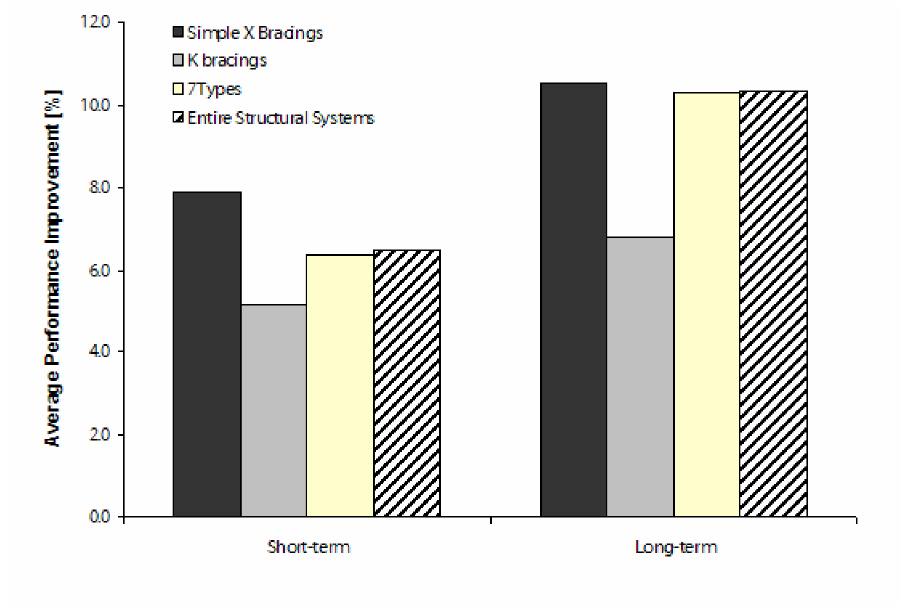


Figure 89. Comparison of the average performance improvements produced in the short- and long-term evolutionary optimization of the entire steel structural systems and wind bracing systems

Even though the performance improvement exceeded 10 percent in the long-term experiments, the best designs produced in the evolutionary optimization processes were substantially inferior to the best designs generated by multiple 1D CAs. Figure 90 graphically illustrates these results and compares them the performance improvements obtained in the evolutionary optimization of wind bracing systems and reported in the previous sections.

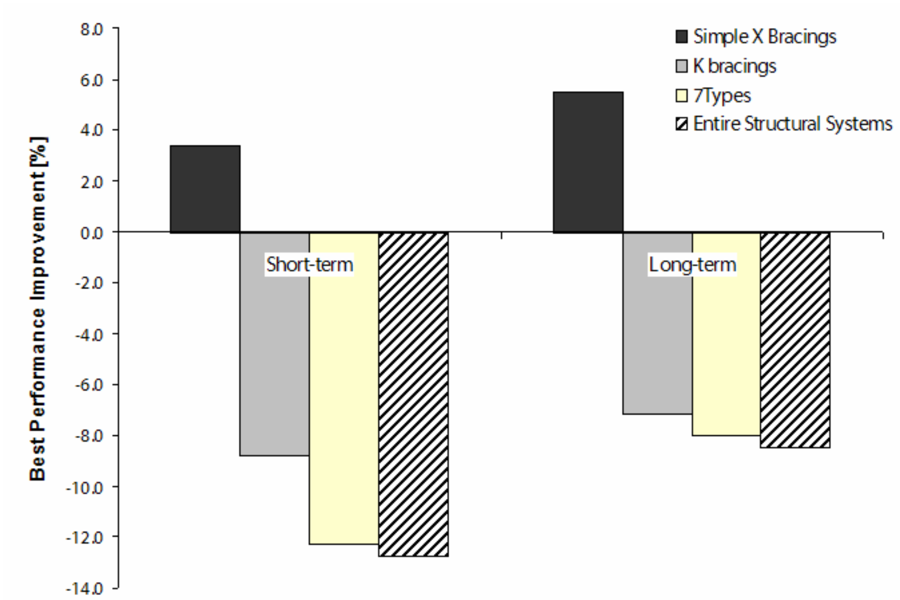
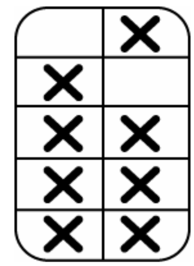


Figure 90. Comparison of the performance improvements between the best designs produced in the evolutionary optimization experiments and the best designs generated by the generative representations base on cellular automata

In the next section, I will empirically investigate whether, or not, the domain knowledge can improve the evolutionary optimization processes. The structural engineering knowledge will be added in a form of a set of structural designs known from the literature, which will be used as initial parents.

7.3.2. Starting from Known Designs

In this section, the impact of applying domain knowledge on the performance of evolutionary optimization processes has been investigated. The structural engineering knowledge was incorporated in the initialization method which utilized a set of designs known from the engineering literature as the initial population of parents. The optimization processes initialized with known designs were subsequently compared with the ones that were initialized randomly.

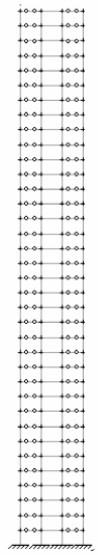
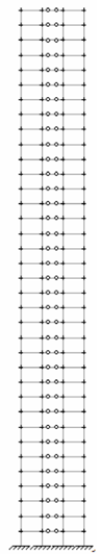
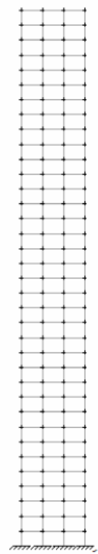
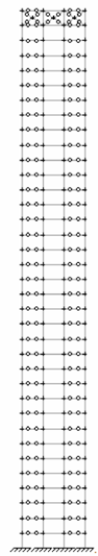
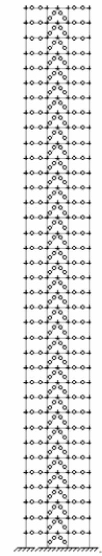
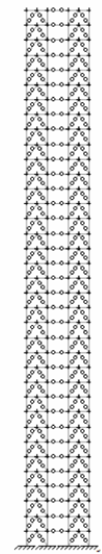
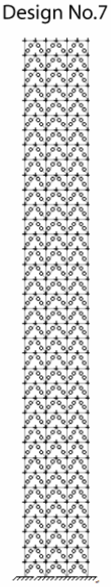
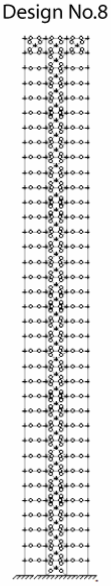
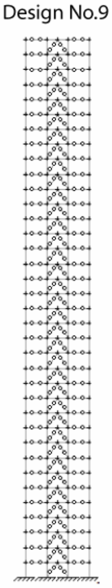
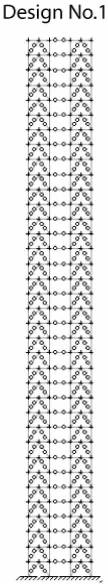
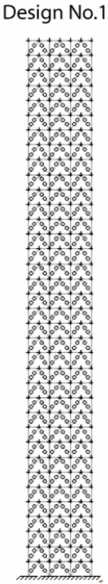
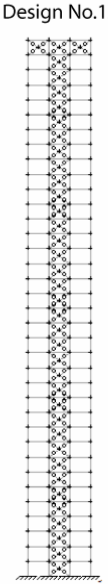


The set of initial parents, shown in Table 75, included 12 designs that were considered as appropriate (called here ‘sub-optimal’) for the class of tall buildings considered in this section, e.g. designs No. 7 and 11, as well as designs that could be characterized as rather inappropriate (called here ‘poor’), e.g. designs No. 1, 2, 4, and 5. The individual designs within the group can be described as:

- Design No.1: one-bay centrally located rigid frame
- Design No.2: two one-bay rigid frames located in outer bays
- Design No.3: three-bay rigid frame
- Design No.4: one-bay centrally located rigid frame with one horizontal truss
- Design No.5: one-bay centrally located rigid frame with one vertical truss
- Design No.6: two one-bay rigid frames located in outer bays with two vertical trusses located in outer bays
- Design No.7: three one-bay vertical trusses
- Design No.8: one-bay centrally located rigid frame with one horizontal truss and centrally located vertical truss
- Design No.9: one-bay centrally located vertical truss
- Design No.10: two one-bay vertical trusses located in outer bays
- Design No.11: three-bay rigid frame with three vertical trusses
- Design No.12: three-bay rigid frame with one horizontal truss

Table 76 shows the problem parameters and their values used in the experiments reported in this section. As it is shown in Table 75, 36-story buildings with 3 bays were studied. For this class of tall buildings, i.e. for buildings with a large value of the aspect ratio, the serviceability conditions play an important role in determining the feasibility of generated design concepts. They constrained the maximum horizontal displacement of structural systems to be no more than $\frac{1}{600}$ of the height of a tall building. These constraints were imposed on the sizing optimization algorithm, implemented in SODA, which adjusted the sizes of all structural members so that the serviceability conditions were satisfied. If a produced design concept did not satisfy the serviceability conditions, it was regarded as infeasible and assigned 0 fitness value (death penalty). As in the previous sections, the fitness of produced design concepts was determined by the total weight of the steel structural systems calculated using the first-order structural analysis. Also, 7 types of wind bracings elements, 2 types of beams, and 2 types of supports were considered.

Table 75. Set of 12 designs known from the structural engineering literature and used as initial parents in the experiments

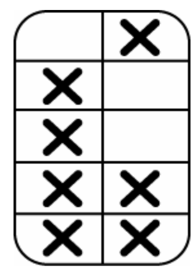
Design No.1	Design No.2	Design No.3	Design No.4	Design No.5	Design No.6
					
3,543,720 3,543,720	3,543,720 3,543,720	1,397,882 3,543,720	3,668,021 3,668,021	4,594,901 4,594,901	5.646,082 5,646,082
48.6806 54.4681	24.8265 26.2494	9.9771 5.8434	20.8332 21.9740	43.9445 48.7332	22.368 23.5535
Design No.7	**Design No.8**	**Design No.9**	**Design No.10**	**Design No.11**	**Design No.12**
					
1,015,753 1,063,923	5,118,195 5,118,195	4,594,901 4,594,901	5,646,082 5,646,082	1,040,868 1,138,524	1,272,675 1,320,391
9.8546 9.8915	18.7918 19.7144	44.2428 49.0780	22.5249 23.7189	10.0713 10.0537	10.0812 10.0798

As before, the design experiments were divided into two groups: the short-term and the long-term optimization processes. Their results are reported in the following subsections.

Table 76. Problem parameters and their values used in the conducted experiments

Problem Parameter	*Value(s)*
Problem type	Design of the entire steel structural system in a tall building
Number of stories	36
Number of bays	3
Bay width	20 feet (6.01 m)
Story height	14 feet (4.27 m)
Distance between transverse systems	20 feet (6.01 m)
Types of bracing elements	No, Diagonal \, Diagonal /, K, V, Simple X, and X
Types of beam elements	Pinned-Pinned, and Fixed-Fixed
Types of column elements	Fixed-Fixed (only)
Types of supports	Pinned, and Fixed

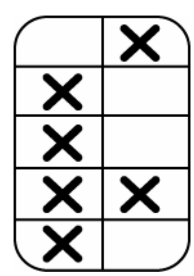
Short-Term Evolutionary Optimization



In this subsection, I investigate short-term evolutionary optimization processes in which, as before, a relatively low budget of 1,000 fitness evaluations per run was used. A sensitivity analysis conducted during the short-term experiments included the following evolutionary computation parameters: mutation and crossover rates, and parent and offspring population sizes. Also, two methods of initialization of evolutionary optimization processes were studied in order to compare the advantages of applying domain knowledge (initialization using a set of known designs) over traditionally used random initialization. Table 77 shows the evolutionary computation parameters used in the experiments reported in this subsection.

First, the short-term optimization processes were started with small populations consisting of 3 designs. Next, a larger population of all 12 parents was used to optimize the steel structural systems in tall buildings. An extensive parameter search of mutation and crossover rates was conducted for all combinations of parent and offspring population sizes to determine their optimal values. The results of these experimental studies were later compared and the optimal experimental setups were used in the long-term experiments.

Small Population Sizes



In this group of experiments, the set of 12 known designs shown in Table 75 was arbitrarily divided into four populations, each of three parents. All populations were then independently evolved. The four populations consisted of the following designs:

- Population No.1: designs No.1, 5, and 9
- Population No.2: designs No.2, 6, and 10
- Population No.3: designs No.3, 7, and 11
- Population No.4: designs No.4, 8, and 12

Table 77. Evolutionary computation parameters used in the short-term optimization experiments

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies
Population sizes (parent, offspring)	(3,15) or (12,60)
Generational model	Overlapping ($\mu+\lambda$)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.025, 0.1, 0.3, or 0.5
Crossover (type, rate)	(uniform, 0), (uniform, 0.2), or (uniform, 0.5)
Fitness	Total weight of the steel structure (determined by the first-order analysis)
Initialization method	Known designs, or random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	1,000 fitness evaluations
Number of runs	15 in each experiment

As discussed earlier, in this chapter parameterized representations of the entire steel structural systems were studied (see section 4.2). For the problem parameters shown in Table 76, the genotypes consisted of 220 genes. 108 genes had seven values (attributes representing types of wind bracings), 108 genes had two values (attributes representing beams), and four genes had two values (attributes representing supports).

Optimal Mutation and Crossover Rates

Typical experimental results are presented in Figure 91 which shows the average best-so-far performance of evolutionary algorithms optimizing population No.1. Specifically, the results of four experiments are presented in which 4 different rates of mutation were used. The uniform crossover rate was kept the same and equal to 0.2. The vertical lines represent 99.9% confidence intervals calculated using Johnson's modified t test.

Figure 91 clearly shows that there exists a pattern regarding the influence of the mutation rate: The higher the mutation rate the faster the optimization progress in the initial stages of evolution (see the zoom in window on the left hand side in Figure 91). Even though the lowest mutation rates produced inferior results in the initial stages of the optimization process, they eventually outperformed higher mutation rates at the end of the run (see the zoom in window on the right in Figure 91). High mutation rates turned out to be too disruptive at the end of the run and finally produced inferior results. The best evolutionary optimization progress in the short-term experiments was obtained when the rate of mutation was equal to 0.1. Thus, the most successful rate was higher than the optimal rate identified in the experiments reported in the previous sections.

On the other hand, the impact of the crossover operator was limited only to an increase, or reduction, of variance of the fitness of the produced design concepts. It did not influence the average progress of the evolutionary optimization processes. The results with smallest variance were obtained when the crossover rate equaled 0.2 whereas the largest variance occurred when the crossover operator was not applied at all (crossover rate 0). The graph showing these findings was, however, omitted.

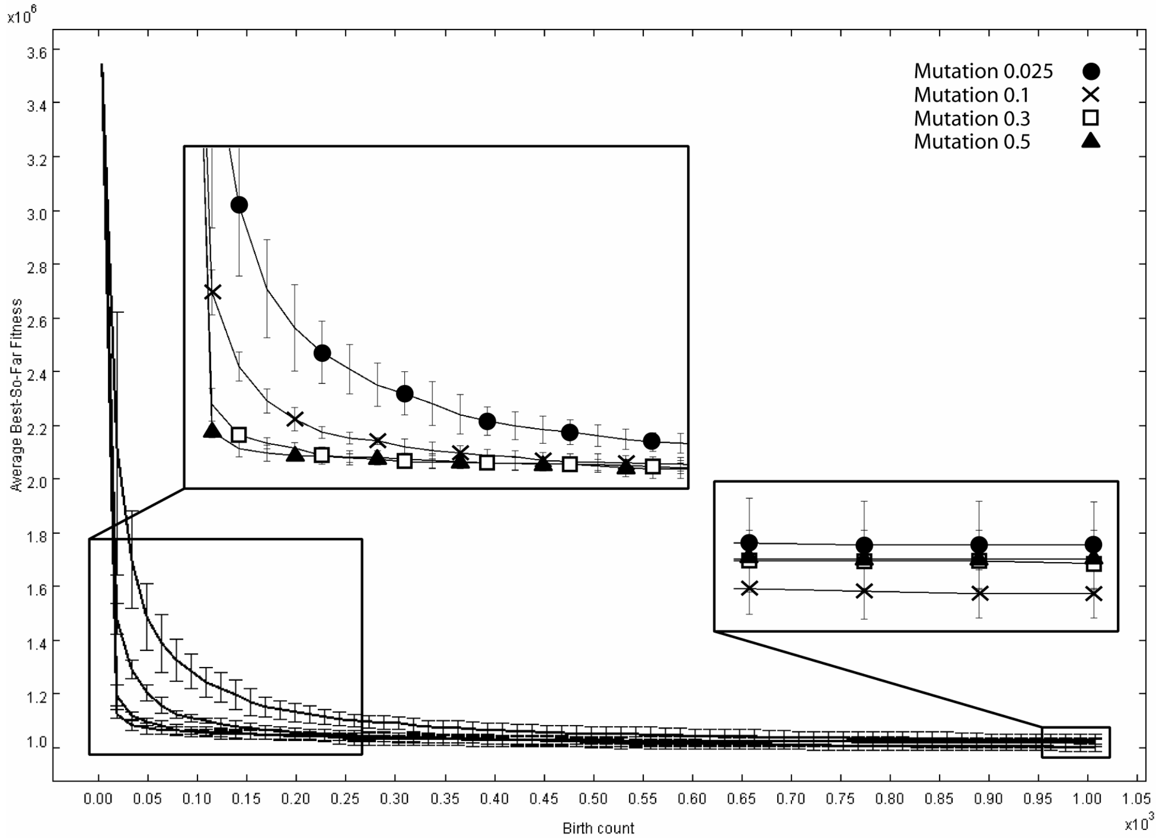


Figure 91. The influence of the mutation rate on the progress of the evolutionary optimization of population No.1

Experimental results have shown that conclusions drawn for population No.1 were also valid for populations No.2 and No.4. However, the situation was different in the case of population No.3 which contained the sub-optimal designs No.7 and 11. Figure 92 shows the impact of the rate of mutation on the progress of evolutionary optimization of population No.3. In this case, high mutation rates, i.e. 0.3 and 0.5, did not produce almost any optimization progress (less than 1 percent in the case of mutation rate equal to 0.5). The best results were obtained when the lowest mutation rate was used, i.e. 0.025. As before, no significant impact of various crossover rates on the average performance of the evolutionary algorithm was observed.

Performance Improvement

The average performance improvements obtained in the design experiments were highly dependent on the fitness of the initial parents. The biggest improvement was achieved when evolutionary optimization processes were started with poor parents, i.e. when populations No.1 and No.2 were used. The smallest optimization progress rates were produced by population No.3 which already contained sub-optimal designs No.7 and 11.

Figure 93 compares the average best-so-far fitness curves obtained in the short-term experiments with 4 populations. Vertical lines in this figure represent 99.9% confidence intervals.

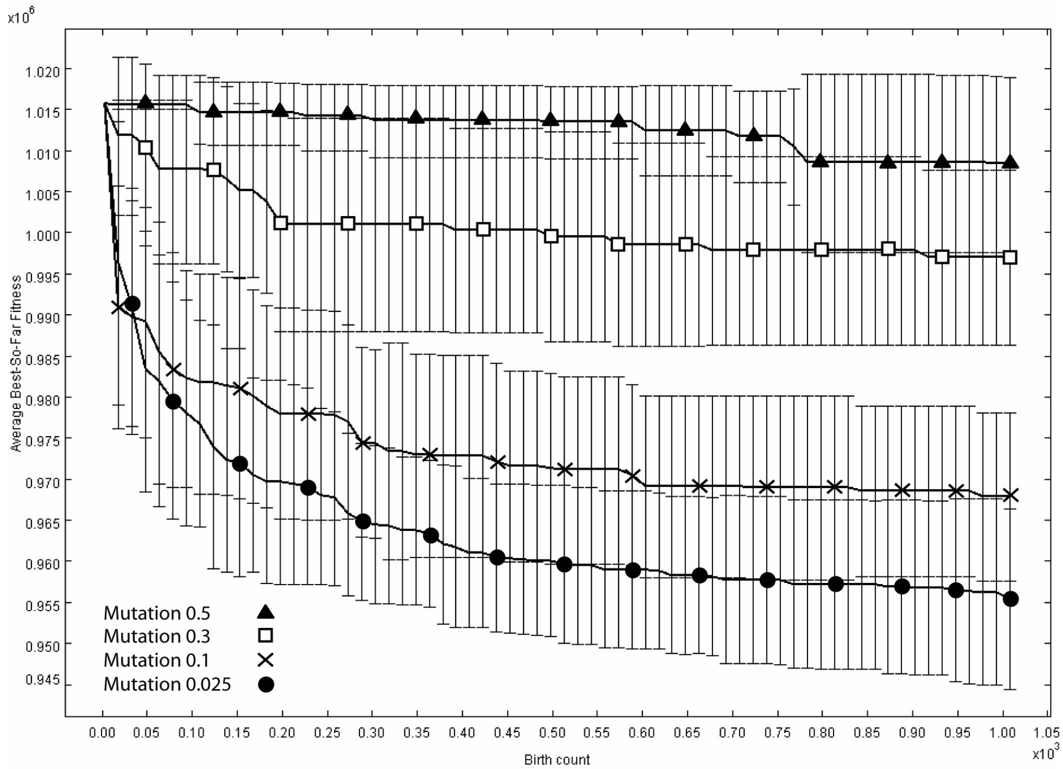


Figure 92. The influence of the mutation rate on the progress of the evolutionary optimization of population No. 3

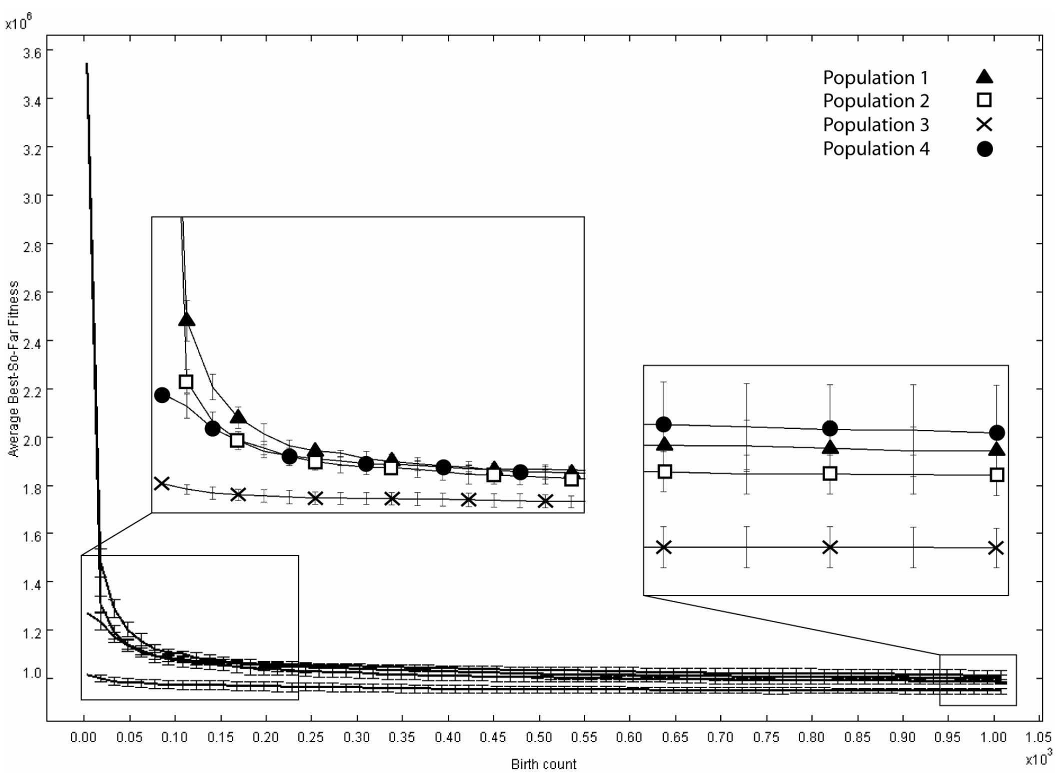


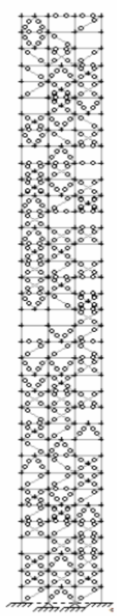
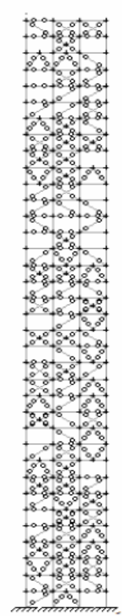
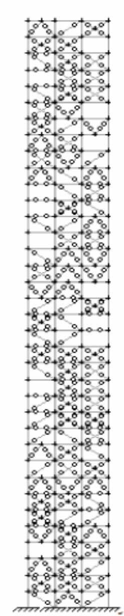
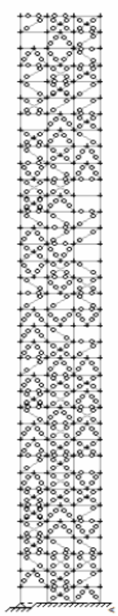
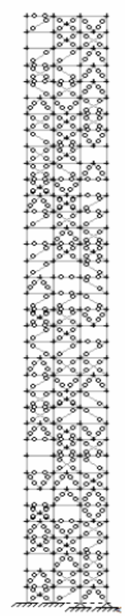
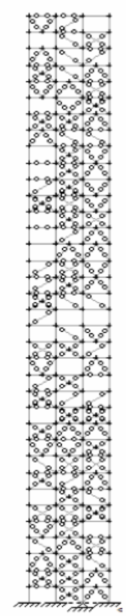
Figure 93. Comparison of the average best-so-far fitness of the entire steel structural systems in tall buildings obtained in the short-term experiments with 4 populations

Figure 93 shows than even though populations No.1 and No.2 achieved the biggest optimization progress during the short-term processes, they did not outperform design concepts generated by population No.3. Optimization of population No.3 produced significantly better end results compared to the other 3 populations. In fact, the end results produced by populations No.1, No.2, and No.4 were similar in terms of the average fitness of the best design concepts.

Optimal Designs

The best design concepts generated by population No.4 are presented in Table 78. As mentioned above, the best design concepts produced by populations No.1 and No.2 were qualitatively and quantitatively similar to the ones shown below. The fitness of the best design concept was equal to 958,189 lbs. and was almost 315,000 lbs., or 25 percent, better than the fitness of the best initial parent (see design No.12 in Table 75).

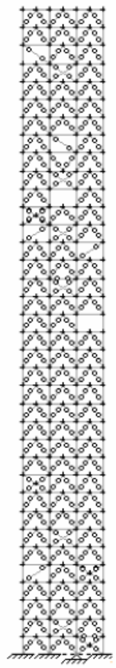
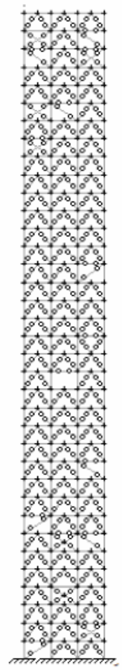
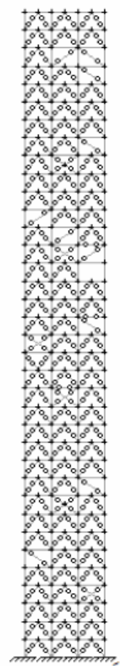
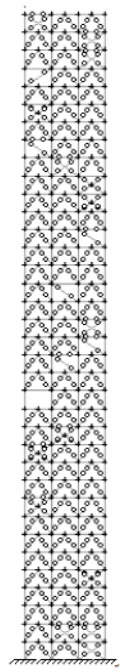
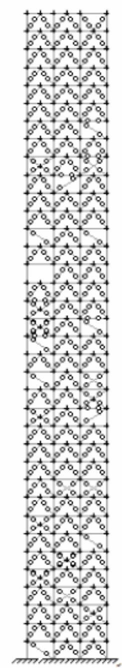
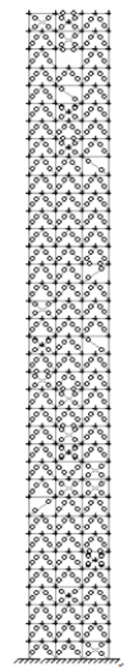
Table 78. Best design concepts produced by population No.4 in the short-term experiments

					
958,189 10.07	966,852 10.03	972,711 9.94	972,930 10.03	978,195 10.02	979,436 10.03

The best design concepts generated by population No.3 are shown in Table 79. They not only outperformed the best design concepts generated by other populations but also exhibited qualitatively different structural shaping patterns. All design concepts in Table 79 show variations of the fully braced pattern composed of K bracings. Occasionally, single K bracings were replaced by other types of wind bracing elements but the overall pattern composed of K bracings could be easily identified. Thus, all best concepts produced during the evolution of population No.3 were restricted to slight mutations of the sub-optimal initial parents (designs No.7 or No.11 in Table 75).

The fitness of the best design concept generated by population No.3 was equal to 933,245 lbs. It was more than 80,000 lbs., or 8 percent, better than the fitness of the best initial parent (design No.7 in Table 75). This performance improvement is more than 3 times smaller than the one achieved by population No.4 but in this case evolutionary optimization processes were initialized with already sub-optimal designs. The average performance improvement of 8 percent for such fit designs constitutes quite a good achievement.

Table 79. Best design concepts produced by population No.3 in the short-term experiments

					
933,245 10.08	933,293 10.08	934,219 10.08	934,287 10.08	934,849 10.08	937,055 10.08

Known Designs vs. Random Initialization

The short-term evolutionary optimization processes started from known designs were also compared to the processes initialized randomly. Figure 94 shows a typical average best-so-far curve obtained for randomly initialized populations and compares it to the corresponding curves (produced using exactly the same parameters) generated by populations No.3 and No.4. It clearly shows that the fitness of randomly initialized parents is, on average, better than the initial parents from population No.4 (and also No.1 and No.2) but worse than the initial parents from population No.3. On the other hand, the end results obtained in the short-term experiments were only slightly better than the end results achieved by population No.4 (statistically insignificant when 95% confidence intervals are considered, see Figure 94). The randomly initialized population did not produce design concepts comparable to the ones generated by population No.3.

The performance improvement achieved by the randomly initialized population in the short-term experiments exceeded 100,000 lbs., or 9 percent. The fitness of the best design concept produced by this population was equal to 967,642 lbs. Thus, it was worse than the best design concepts produced by both population No.4 (see Table 78) and population No.3 (see Table 79).

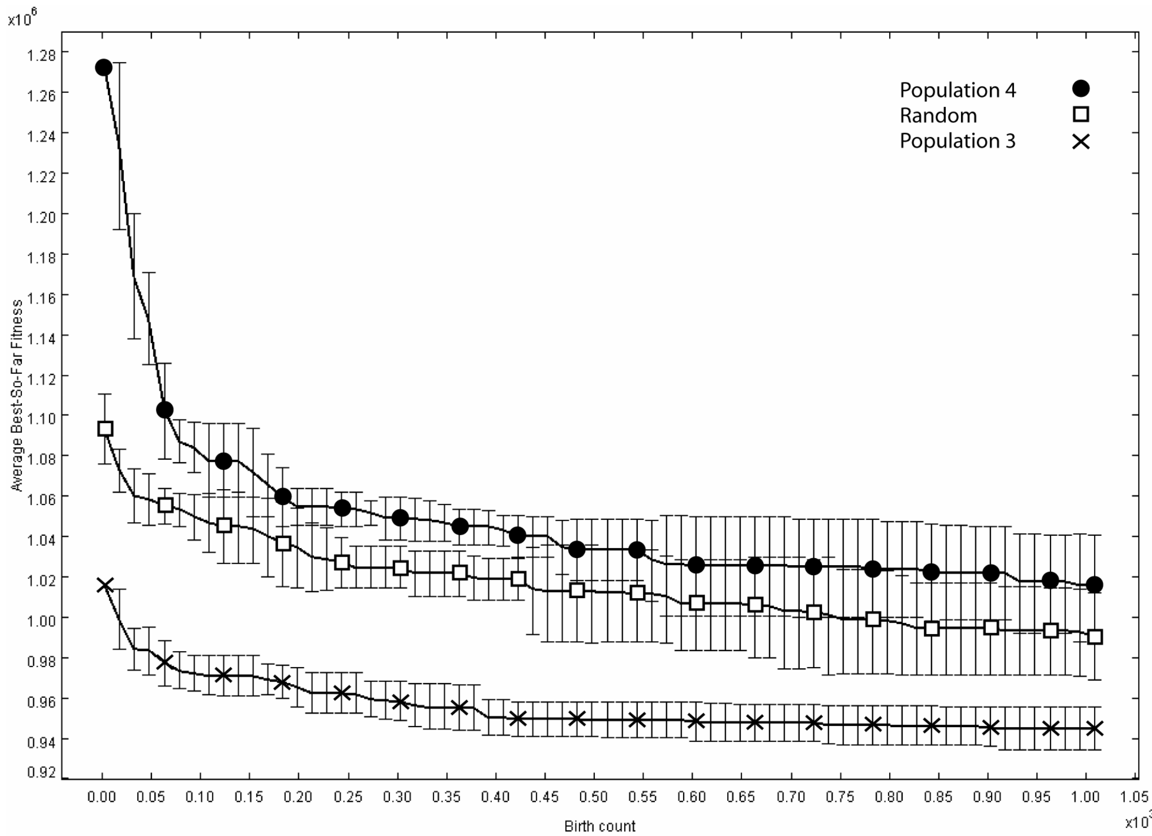
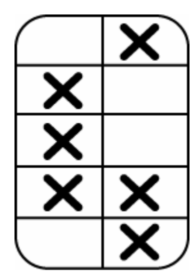


Figure 94. Comparison of the average best-so-far fitness of the entire steel structural systems obtained in the short-term experiments with populations No.3 and No.4 and a population initialized randomly

In the following subsection, I will investigate the impact of increasing the size of the parent population on the fitness of produced design concepts.

Large Population Sizes



In the design experiments reported in this subsection, the entire set of 12 known designs was employed is a single large population of initial parents. Also, in order to compare the impact of the initialization method on the fitness of produced design concepts, another group of design experiments was conducted in which exactly the same parameters were used but the initial population of parents was generated randomly.

Optimal Mutation and Crossover Rates

A sensitivity analysis involving mutation and crossover rates revealed that the same mutation rates that worked well in the case of population No.3 also generated the best results for the large population initialized with the entire set of 12 known designs. The best evolutionary optimization progress was obtained when the rate of mutation equaled 0.025. As before, various crossover rates did not influence the fitness of the produced design concepts. Graphs showing these results have been, however, omitted.

Large vs. Small Population Sizes

Figure 95 compares the average best-so-far curves obtained in the experiments with the large population initialized with known designs, the large population initialized randomly, and population No.3. It shows that both small and large populations initialized with known design concepts significantly outperform the large population which was initialized randomly. The differences in the optimization progress between population No.3 and the large population occur mainly in the initial stages of evolution. The end results, however, are similar and the differences between average fitnesses are statistically insignificant. In fact, the average fitness of the best design concepts generated by the large population slightly outperformed the value produced by population No.3. Similar relationships have been identified between small and large populations initialized randomly.

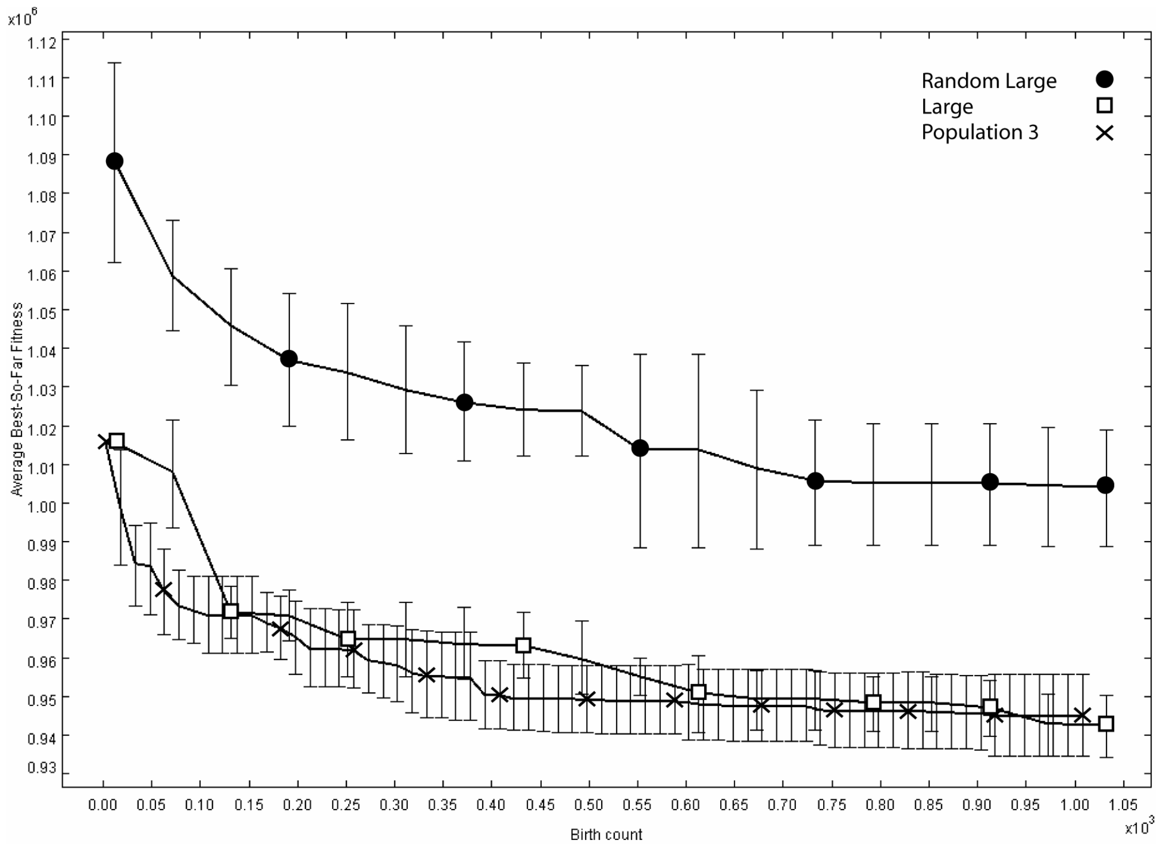


Figure 95. Comparison of the average best-so-far performance obtained in the short-term experiments with the large population initialized with known design concepts, the large population initialized randomly, and population No.3

Optimal Designs

The best design concepts produced during the evolutionary optimization of the large population initialized with known designs are presented in Table 80. As it was the case with the best design concepts generated by population No.3 (see Table 79), all designs shown in Table 80 exhibit various mutations of the fully-braced pattern composed of K bracings. They were qualitatively different than the design concepts produced during the evolutionary optimization of the large population initialized randomly (see Table 81).

Table 80. Best design concepts produced in the short-term design experiments by the large population initialized with 12 known designs

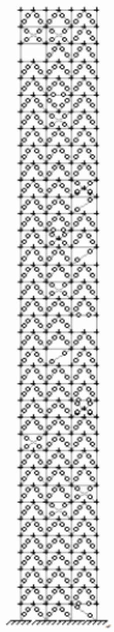
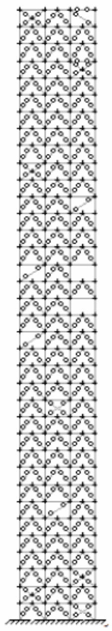
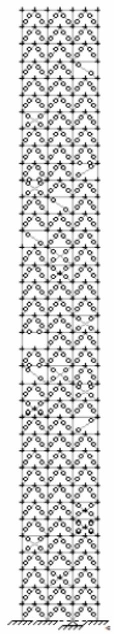
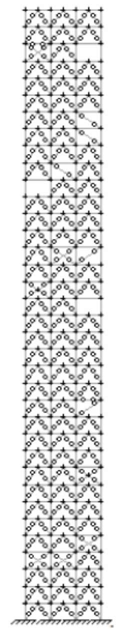
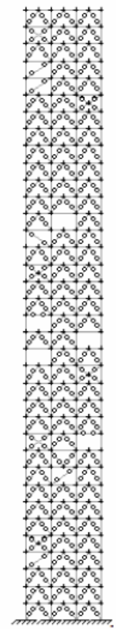
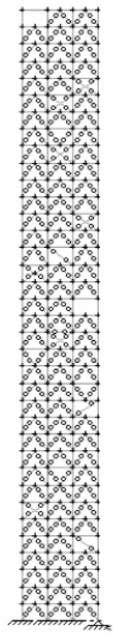
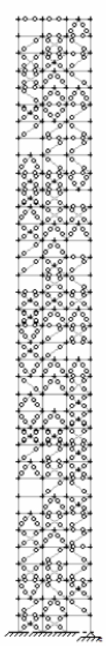
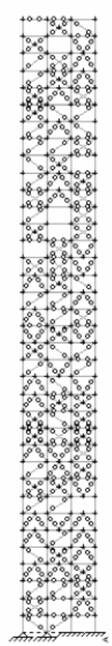
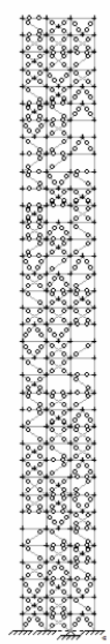
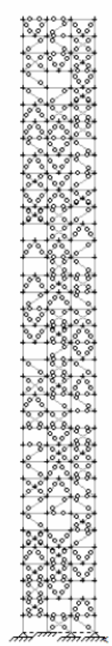
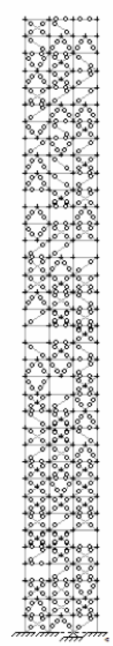
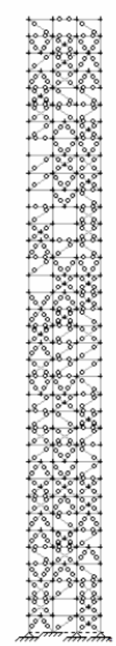
					
932,216 10.07	938,587 10.08	939,160 10.08	939,893 10.07	940,518 10.08	940,630 10.07

Table 81. Best design concepts produced in the short-term design experiments with a large population of randomly generated initial parents

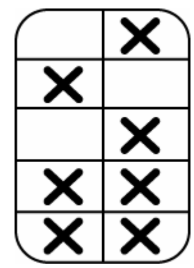
					
972,534 10.04	974,283 10.03	974,682 10.08	977,802 10.08	979,328 10.08	980,908 10.07

Performance Improvement

The performance improvements achieved in the short-term experiments with the population initialized with known designs and the population initialized randomly exceeded 8 percent and 9.5 percent, respectively. The fitness of the best design concept produced by the former population was equal to 932,216 lbs. and was slightly better than the fitness of the best design concept produced by population No.3 (see Table 79).

In the next subsection, I will describe results of the long-term evolutionary optimization experiments with populations initialized with known design concepts. In this group of experiments, the optimal evolutionary computation parameters, described above, were employed.

Long-Term Evolutionary Optimization



In this subsection, I describe results of the long-term design experiments involving both small and large populations initialized with known design concepts. Next, I compare them with the results produced by evolutionary optimization processes with exactly the same parameters but initialized randomly.

Table 82 shows the evolutionary computation parameters that were used in the experiments reported in this section. The most successful rates of mutation and crossover, identified in the short-term experiments, were employed here in the long-term optimization processes.

In the first group of experiments, all populations (both small and large) were evolved with the mutation rate equal to 0.1 (this rate was identified as optimal for populations No.1, No.2, No.4 and the populations initialized randomly). In the second group of experiments, evolutionary optimization experiments were repeated for population No. 3 and the large population initialized with known designs, this time using the mutation rate of 0.025 (previously identified as the optimal rate for population No.3 and the large population).

Optimal Mutation and Crossover Rates

Figure 96 compares average fitness values obtained at the end of the short-term and long-term experiments for all populations investigated in this section. Vertical lines in this figure represent 95% confidence intervals. The average fitness values and confidence intervals corresponding to the short-term processes were taken from the experiments in which optimal crossover and mutation rates were employed for each population. On the other hand, the results corresponding to the long-term processes come from experiments in which the rates of mutation and crossover were uniform across all populations and equal to 0.1 and 0.2, respectively.

Figure 96 illustrates several interesting phenomena which occurred in this group of experiments. First, all populations, for which the optimal values of mutation and crossover values were employed, produced significantly better results in the long-term experiments than in the short ones. However, population No.3 and the large population initialized with known designs produced worse results in the long-term experiments than in the short-term processes.

For these two populations, the optimal rate of mutation was equal to 0.025. Thus, by using inappropriate rates of genetic operators in evolutionary optimization experiments, inferior results might be obtained even when significantly longer evolutionary optimization processes are conducted.

Table 82. Evolutionary computation parameters and their values used in the long-term optimization experiments of the entire steel structural systems in tall buildings

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies
Population sizes (parent, offspring)	(3,15) or (12,60)
Generational model	Overlapping (μ+λ)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.1 for all populations
	0.025 for population No.3 and the large population
Crossover (type, rate)	(uniform, 0.2)
Fitness	Total weight of the steel structure (determined by the first-order analysis)
Initialization method	Known designs, or random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	10,000 fitness evaluations
Number of runs	5 in each experiment

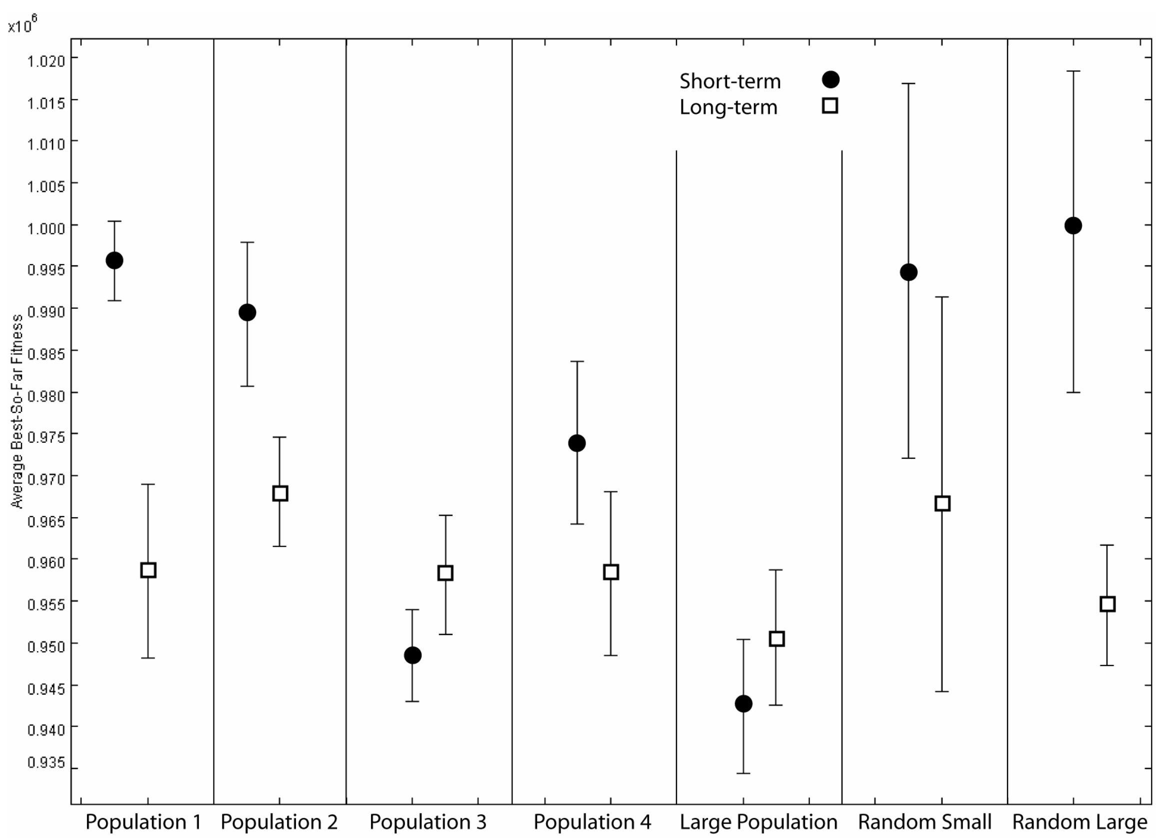


Figure 96. Comparison of the average best-so-far fitness obtained in short-term and long-term experiments

Finally, Figure 96 clearly shows dissimilarities between variances obtained using the two initialization methods. The variance of the obtained results was substantially larger when populations had been initialized randomly. However, as the rightmost part of Figure 96 shows, it was significantly reduced in the long-term evolutionary optimization processes involving the randomly initialized population of large size.

Figure 97 illustrates the impact of the rate of mutation on the performance of the evolutionary optimization of population No.3 and the large population. It clearly shows that when optimal values of the mutation rate are used, i.e. 0.025, the produced results are significantly better than in the case when inappropriate rate is applied (here 0.1).

Performance Improvement

The average performance improvements in the long-term experiments were equal to 22,600 lbs. (2.3 percent) and 21,900 lbs. (2.3 percent) for population No.3 and the large population, respectively. Figure 97 also shows that large population produced better results than population No.3 for both mutation rates. It outperformed population No.3, on average, by almost 7,000 lbs., or 0.7 percent, and at the same time produced smaller variance. This is different from the results obtained in the evolutionary optimization of wind bracing systems (see section 7.2) where smaller population sizes proved to be more efficient.

Optimal Designs

The best design concepts produced in the long-term experiments are presented in Table 83. 5 out of 6 design concepts in this table were produced by the large population initialized with known design concepts while the remaining one was generated by population No.3. The fitness of the best design concept was equal to 926,268 lbs. and was almost 90,000 lbs., or 8.8 percent, better than the fitness of the sub-optimal initial parent, i.e. design No.7 in Table 75.

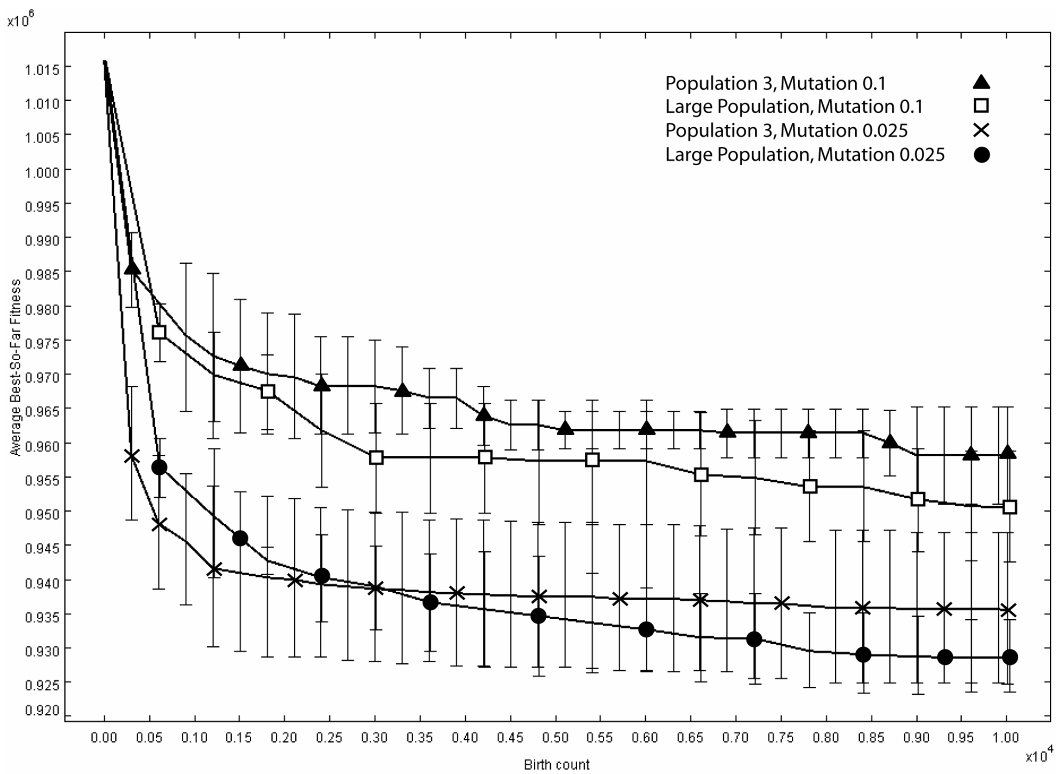
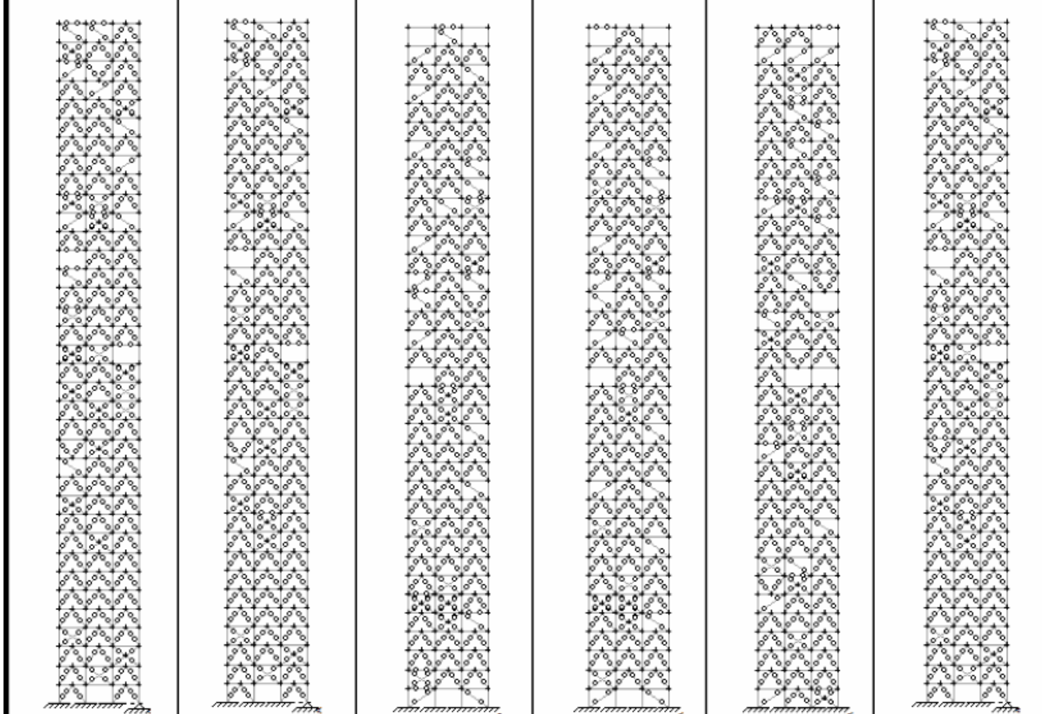


Figure 97. The impact of the mutation rate of the performance of the long-term evolutionary optimization processes for population No.3 and the large population

None of the populations initialized randomly produced design concepts of comparable performance. The fitness of the best design concept produced by the large population initialized randomly was equal to 947,859 lbs. and was more than 21,500 lbs., or 2.3 percent, worse than the overall best design concept found in the long-term experiments (see Table 83).

Table 83. Best design concepts of the entire steel structural systems produced in the long-term experiments



926,269 10.08	926,307 10.08	926,389 10.08	926,698 10.08	926,881 10.08	927,143 10.08

Summary

In this section, I investigated the impact of the domain knowledge on the performance of evolutionary optimization processes. The structural engineering knowledge was applied in a form of a set of 12 known designs of the entire steel structural systems in tall buildings which were used as the initial parents. As in the previous sections, short-term and long-term design experiments were conducted and the impact of selected evolutionary computation parameters was tested empirically in the sensitivity analysis phase.

First, the short-term evolutionary optimization processes were investigated with both small and large population sizes. The experiments have shown that the optimal rates of mutation which produced the best optimization progress are related to the quality of initial parents. When poor design concepts are used in the initial population of parents then higher mutation rates, i.e. 0.1, are preferred. On the other hand, if the initial population contains a highly fit design concept then significantly lower mutation rates produce better results. The results were consistent for both small and large population sizes. The long-term experiments have also shown that the optimal rates of genetic operators should be carefully determined during the short-term processes. Otherwise, when inappropriate rates are used the results produced by the long-term optimization processes may be inferior to the results obtained in much shorter optimization processes.

Evolutionary optimization processes started from poor design concepts achieved superior average performance improvements. However, the overall best design concepts were produced by populations that contained the sub-optimal design concepts as initial parents. Also, evolutionary algorithms with larger population sizes proved to perform better in this problem domain. These findings are illustrated in Figure 98 and Figure 99. Figure 98 shows a comparison of the average performance improvements produced in the short-term and long-term experiments by all populations considered in this section while Figure 99 compares the fitness of the best design concepts produced by these populations.

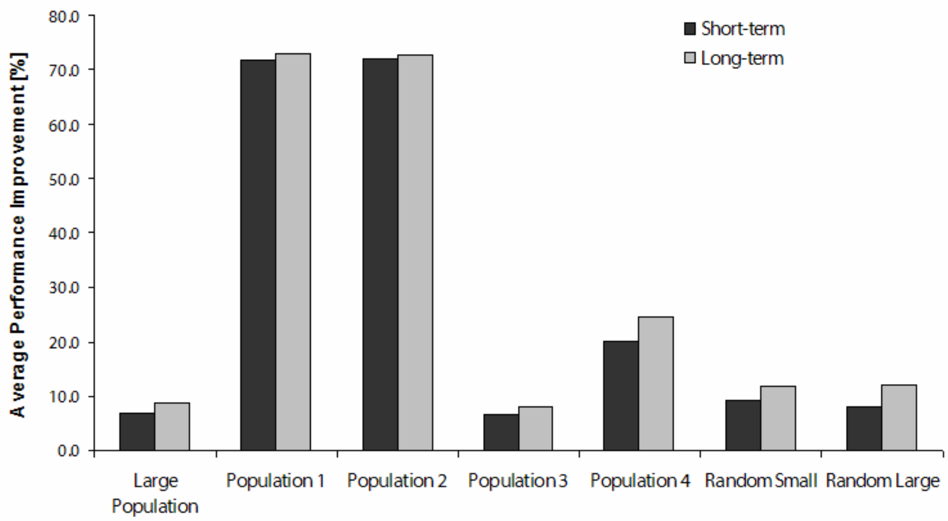


Figure 98. Comparison of the average performance improvements achieved in the evolutionary optimization of the entire steel structural systems in the short-term and long-term experiments

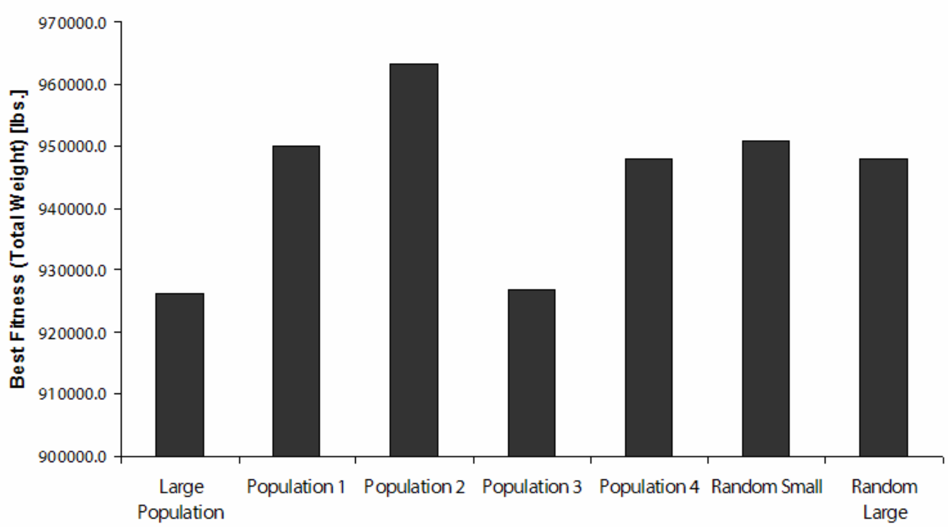
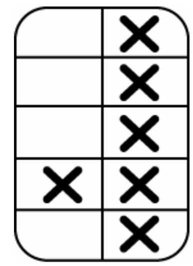


Figure 99. Comparison of the fitness of the best design concepts produced in the long-term optimization experiments by all populations considered in this section (lower values correspond to better designs)

In the next section, I will extend the fitness evaluation from single-objective models to multiobjective approaches. I will achieve it by including the maximum horizontal displacement of the structural system as the second objective with respect to which the design concepts are optimized.

7.4. Multiobjective Optimization of the Entire Steel Structural Systems



So far, evolutionary optimization processes considered only one objective, namely the total weight of a structural system. The second performance measure, i.e. the maximum horizontal displacement, was either only monitored (see sections 7.2 and 7.3.1) or treated as a constraint for structural systems with high aspect ratios (see section 7.3.2). In this section, I investigate more general evaluation models in which both performance measures are considered as objectives with respect to which the entire steel structural systems are optimized.

A simple multiobjective evolutionary algorithm based on aggregating functions (see section 2.1.5) was used in the experiments reported here. Both objectives were combined into a single fitness function using a set of arbitrarily assigned weights which determined the relative importance of each of the two objectives. By considering several combinations of the weights I have attempted to identify the changes of the optimal topologies of steel structural systems when the importance of each of the two objectives was modified. I also tried to determine the approximate shape of the Pareto front in this two-objective performance space.

The problem parameters and their values used in the experiments reported in this section were exactly the same as the ones used in section 7.3.2. They are presented in Table 76. As in section 7.3.2, two methods of initialization of multiobjective evolutionary optimization were considered: a random initialization and an initialization using a set of known designs. In this section, only the long-term multiobjective evolutionary optimization experiments were conducted. The values of evolutionary computation parameters, i.e. population sizes and mutation and crossover rates, were assumed based on the results of the short-term experiments reported in section 7.3.2.

Table 84 shows the evolutionary computation parameters and their values used in the long-term multiobjective evolutionary optimization experiments.

As discussed earlier, the fitness of a design concept was calculated as a weighted average of the normalized total weight of a structural system and the related normalized maximum horizontal displacement. 6 combinations of weighting coefficients were used in the multiobjective design experiments, including 0.0·W+1.0·D, 0.2·W+0.8·D, etc., where W denotes the total weight of the structural system and D its maximum horizontal displacement. Each design concept was represented by a fixed-length genome consisting of 220 genes. 108 genes encoded attributes defining types of wind bracing elements. These genes had 7 possible values representing 7 types of wind bracing elements. 108 genes encoded attributes representing beams. These genes had binary values. Finally, 4 genes encoded types of supports and also had binary values.

The experimental results are presented in the following subsections.

Table 84. Evolutionary computation parameters and their values used in the multiobjective evolutionary optimization experiments

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies
Population sizes (parent, offspring)	(12,60)
Generational model	Overlapping ($\mu+\lambda$)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.1
Crossover (type, rate)	(uniform, 0.2)
Fitness	Weighted average involving two objectives: the total weight of the structural system the maximum horizontal displacement of the structural system (‘sway’)
Weighting coefficients	0.0, 0.2, 0.4, 0.6, 0.8, or 1.0
Initialization method	Known designs, or random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	10,000 fitness evaluations
Number of runs	5 in each experiment

7.4.1. Impact of the Initialization Method

Figure 100 shows two normalized average best-so-far fitness curves obtained in the multi-objective evolutionary optimization experiments with randomly initialized parents and known designs used as initial parents. The vertical lines represent 95% confidence intervals calculated using the modified Johnson’s t test. In this case, the fitness of the design concepts was calculated using the following coefficients: 0.2 for the total weight and 0.8 for the maximum displacement. Figure 100 clearly shows that in this case the evolutionary processes initialized with known design concepts outperformed the ones initialized randomly. However, evolutionary optimization processes initialized with known parents did not always produce superior results.

Figure 101 shows the normalized average best-so-far curves for another combination of weighting coefficients. Here, both multiobjective evolutionary design processes produced similar results. In general, the following pattern was observed in the conducted experiments. For low values of the weighting coefficient associated with the total weight of the steel structural system, the evolutionary optimization processes initialized with known design concepts significantly outperformed the ones initialized randomly. However, when the value of this coefficient was increased (and the value of the coefficient associated with the maximum displacement was decreased) then both initialization methods produced similar results. In some cases, random initialization slightly outperformed the initialization with known designs.

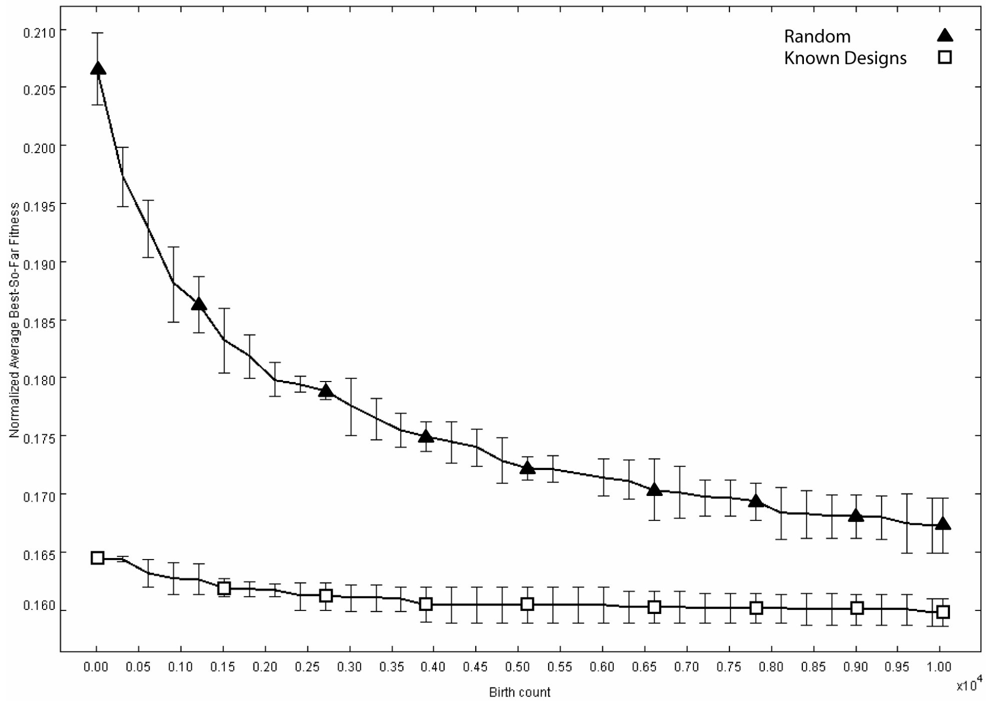


Figure 100. Comparison of the normalized average best-so-far fitness obtained in the multiobjective evolutionary optimization experiments with randomly initialized parents and known designs used as the initial parents (here the fitness was calculated using the formula: 0.2W+0.8D)

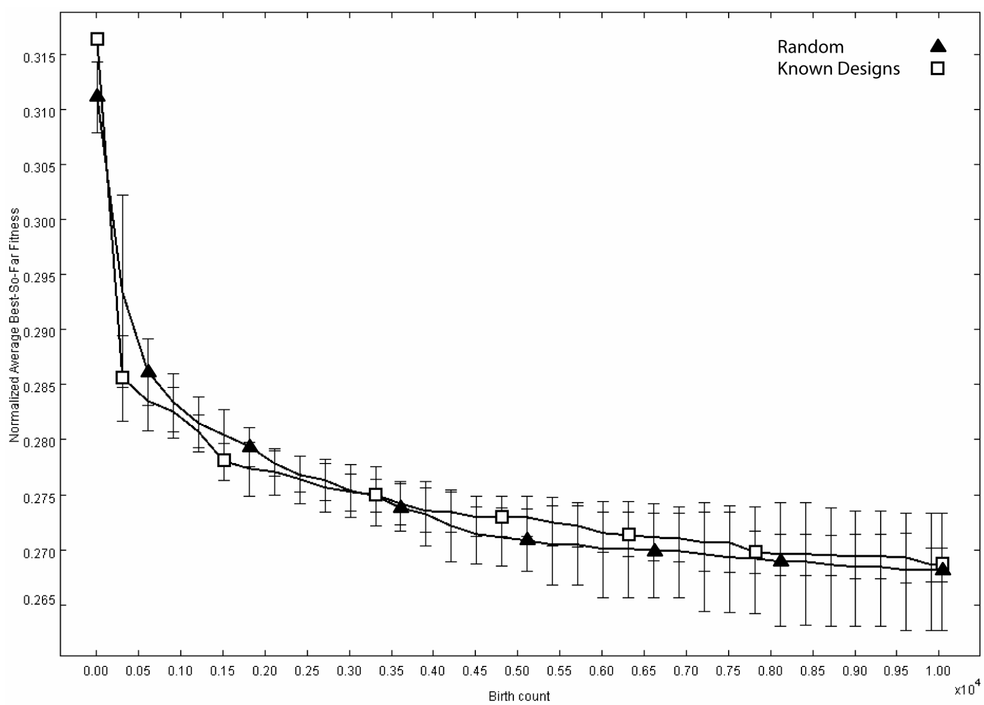


Figure 101. Comparison of the normalized average best-so-far fitness obtained in the multiobjective evolutionary optimization experiments with randomly initialized parents and known designs used as the initial parents (here the fitness was calculated using the formula: 0.6W+0.4D)

7.4.2. Approximate Shape of the Pareto Front

The best design concepts of steel structural systems produced in all design experiments involving various combinations of weighting coefficients were analyzed with respect to the values of both objectives. The results of this analysis are presented in Figure 102.

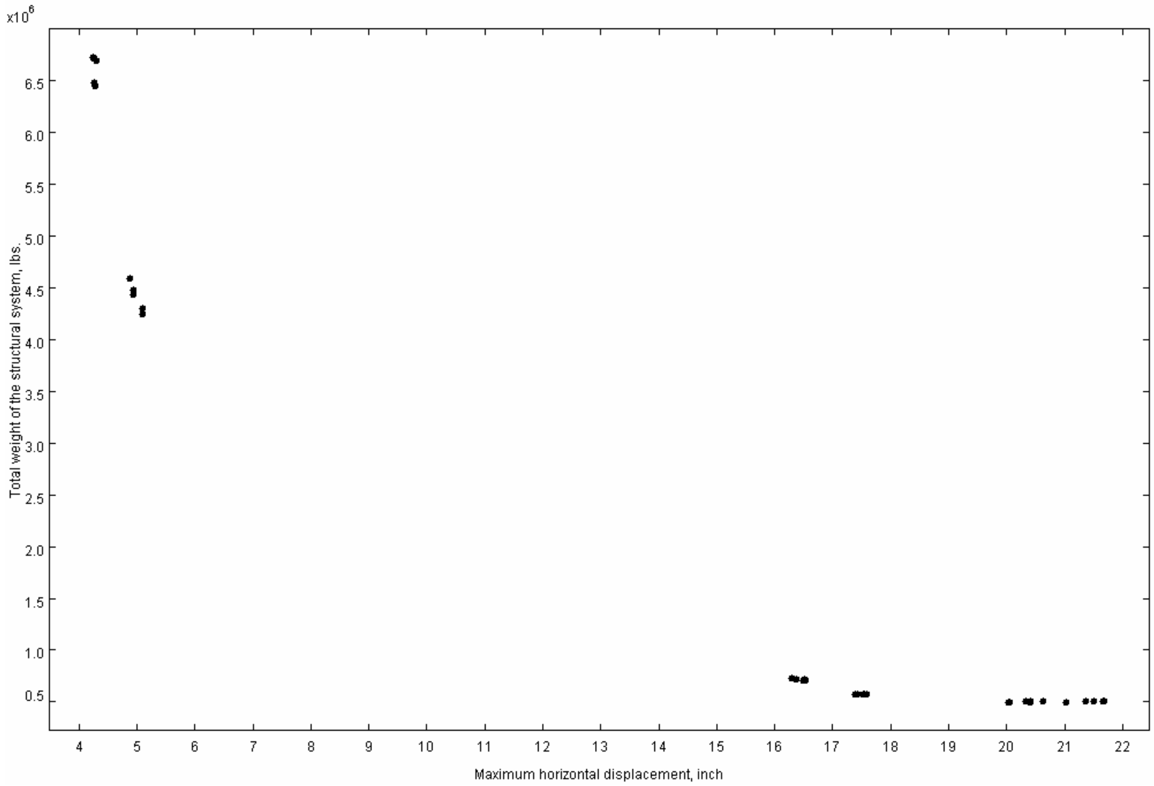


Figure 102. Approximate shape of the Pareto front in the performance space spanned over of the total weight of the steel structural system and its maximum horizontal displacement

It shows an approximate shape of the Pareto front spanned over the performance space formed by the total weight of the structural system and its maximum horizontal displacement. It clearly shows that the total weight of the optimal structural designs varied from about 500,000 lbs. to more than 6,500,000 lbs. At the same time, the maximum horizontal displacements of the structural systems ranged from 4 inches to almost 22 inches. Figure 102 also shows that there is a strong trade-off between the two objectives.

7.4.3. Optimal Topologies of Steel Structural Systems

The best design concepts shown in Figure 102 were also analyzed qualitatively for changes in their topologies occurring when the importance of each of the two objectives was modified. Figure 103 shows the topologies of the structural systems associated with the approximate Pareto front which was discussed in the previous subsection.

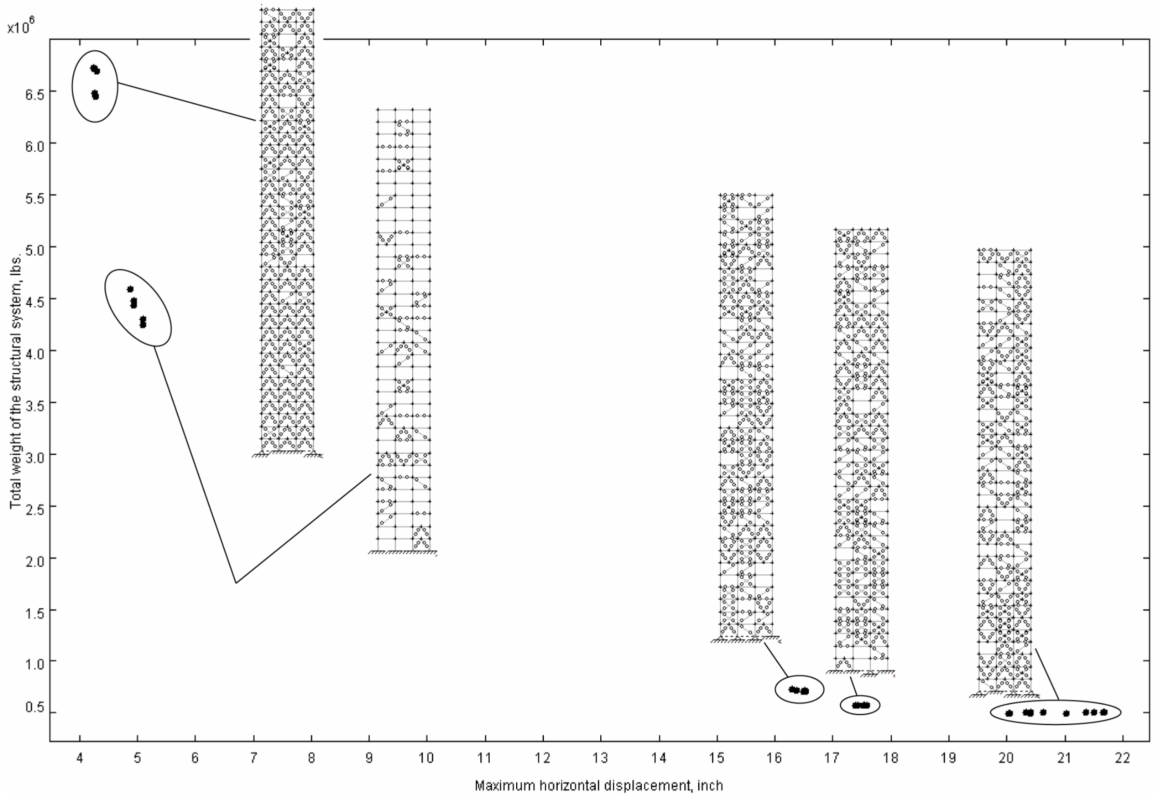


Figure 103. Topologies of the optimal structural systems associated with various regions of the approximate Pareto front

Figure 103 clearly shows that there are significant qualitative differences among the topologies of structural systems located in various parts of the Pareto front. The leftmost design, which corresponds to the region of the Pareto front with the smallest horizontal displacements and the largest total weight, exhibits a dramatically different structural shaping pattern than the second design shown to the left. In the former case, a fairly uniform pattern of K bracings can be identified with occasional occurrences of X bracings. In the latter case, wind bracing elements appear only occasionally and the stiffness of the structural system is provided by the increased cross-sections of beams and columns. The three rightmost designs in Figure 103 are again different than the previously described designs. Here, combinations of relatively large numbers of X and K bracings can be identified. The topologies of the three rightmost design concepts are much more similar than the leftmost designs.

When we compare the design concepts shown in Figure 103 to the ones generated in the design experiments in which the total weight of the structural system was used as the only objective and the maximum horizontal displacement was imposed as a constraint (see Table 83), we can identify significant qualitative and quantitative differences. The designs shown in Table 83 are almost 50% heavier than the rightmost designs shown in Figure 103. At the same time they exhibit substantially smaller (also about 50%) horizontal displacements.

The quantitative characteristics of the structural systems shown in Table 83, i.e. their total weights and the maximum horizontal displacements, show that these designs are located close to the central region of the Pareto front. If we use this information we can more accurately approximate the shape of the Pareto front in this two-objective performance space, as it is shown in Figure 104.

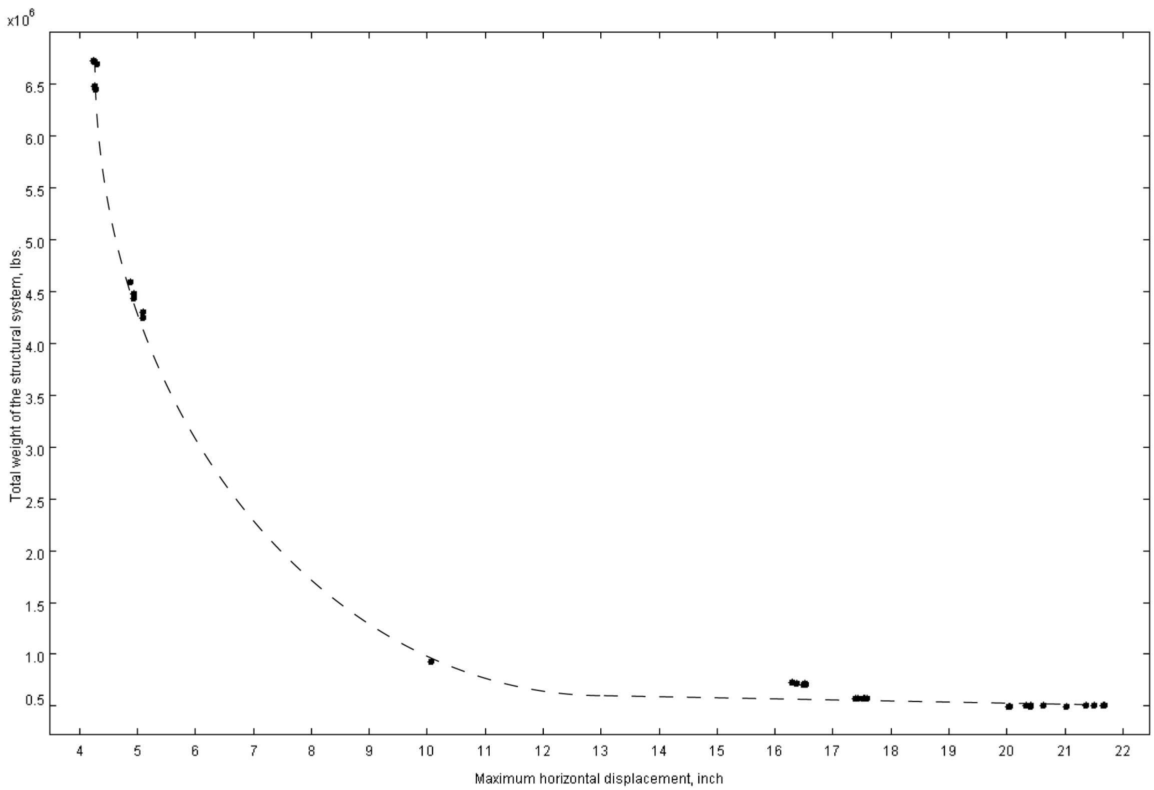


Figure 104. More accurate approximation of the shape of the Pareto front in the performance space spanned over of the total weight of the steel structural system and its maximum horizontal displacement

7.4.4. Summary

In this section, I studied multiobjective evolutionary optimization of the entire steel structural systems in tall buildings. A simple multiobjective evolutionary algorithm based on aggregating functions was used in these experiments. In the experiments, two objectives were considered: the total weight of a structural system and its maximum horizontal displacement. Both objectives were combined into a single fitness function using a set of arbitrarily assigned weights which determined the relative importance of each of the two objectives.

I identified the approximate shape of the Pareto front in this two-objective performance space by considering several combinations of weighting coefficients. The conducted analysis has shown that there is a strong trade-off between the two objectives. I also found that the topologies of the steel structural systems corresponding to various regions of the Pareto front exhibit quite different structural shaping patterns.

The conducted experiments provided new insights on the ranges of variability of the two objectives in which the optimal design concepts of the entire steel structural systems can be found. They also provided a broader understanding of this complex structural design problem by identifying the optimal topologies for various regions of the Pareto front.

7.5. Summary

In this chapter, I described the results of the second stage of the Empirical Performance Validation process (see section 3.6.3) in which I investigated the evolutionary computation component of Emergent Engineering Design. I have attempted to build confidence in the usefulness of this component of EED by presenting and discussing the results of a large number of evolutionary optimization experiments.

In the first section of this chapter, I discussed criteria of optimality of steel structural systems in tall buildings and revisited the research question 3 and the research hypothesis 3. I also refined them in the context of design problems considered in this dissertation.

In the second section of this chapter, I described results of the design experiments in which wind bracing systems in tall buildings were optimized. I investigated the impact of various evolutionary computation parameters on the performance of the evolutionary algorithm in this problem domain and defined optimal values of these parameters. I also reported that evolutionary algorithms achieved significant performance improvements (more than 10 percent) in the long-term optimization experiments. In some cases, however, they produced substantially inferior designs to the ones generated by the generative representations (see the results reported in chapter 6).

In the third section of this chapter, I investigated evolutionary optimization of a more complex design problem, namely conceptual design of the entire steel structural systems in tall buildings. I empirically showed that evolutionary algorithms performed well in this complex problem domain and achieved significant performance improvements which again exceeded 10 percent. I also described how one can add domain knowledge to the evolutionary optimization processes in a form a set of known design concepts used as the initial parents. I demonstrated that adding this knowledge improves the performance of the evolutionary optimization processes.

In the fourth section of this chapter, I extended the traditional single-objective evaluation models and investigated the multiobjective evolutionary optimization of the entire steel structural systems in tall buildings in which the design concepts were minimized with respect to both the total weight and the maximum horizontal displacement. I empirically showed the approximate shape of the Pareto front. It provided a 'big picture' of this two-objective performance space. I also found that there were strong trade-offs between the two objectives. Furthermore, I demonstrated how the optimal topologies of the steel structural systems change in various regions of the Pareto front.

In the next chapter, I will empirically investigate integrated components of EED, i.e. the generative representations combined with evolutionary algorithms.

8. MORPHOGENIC EVOLUTIONARY DESIGN

> *"The most extensive computation known has been conducted over the last billion years on a planet-wide scale: it is the evolution of life."*
>
> *(David Rogers)*

In this chapter, I discuss results of the design experiments in which integrated components of Emergent Engineering Design, i.e. the generative representations component and the evolutionary computation component, were tested to effectively generate novel design concepts of steel structural systems in tall buildings and to efficiently optimize them. The results presented here constitute the third stage of the Empirical Performance Validation process, as discussed earlier in section 3.6.3. Similarly as before, the design experiments reported in this chapter have been conducted using Emergent Designer.

Figure 105 shows the organization of this chapter. First, in section 8.1, I revisit the fundamental research question and the fundamental research hypothesis of this dissertation and refine them in the context of the considered design problems. I also provide an overview of the morphogenic evolutionary design experiments reported in this chapter.

Next, section 8.2 describes the results of the morphogenic evolutionary design of wind bracing systems in tall buildings. The experiments reported in this section were divided into three groups:

1. Experiments in which the generative representations of wind bracing systems based on elementary CAs were evolved using evolutionary algorithms (see subsection 8.2.1).
2. Experiments with one-dimensional CAs with 7 cell values representing 7 types of wind bracings elements evolved by evolutionary algorithms (see subsection 8.2.2).
3. Experiments in which generative representations based on two-dimensional CAs were evolved (see subsection 8.2.3).

Finally, the experimental results of the morphogenic evolutionary design of the entire steel structural systems in tall buildings are reported in section 8.3. Here, the generative representations of all subsystems of the steel structural system in a tall building, i.e. the wind bracing subsystem, the beam subsystem, and the supports, were evolved by evolutionary algorithms. The generative representations investigated in this section were based on multiple one-dimensional CAs.

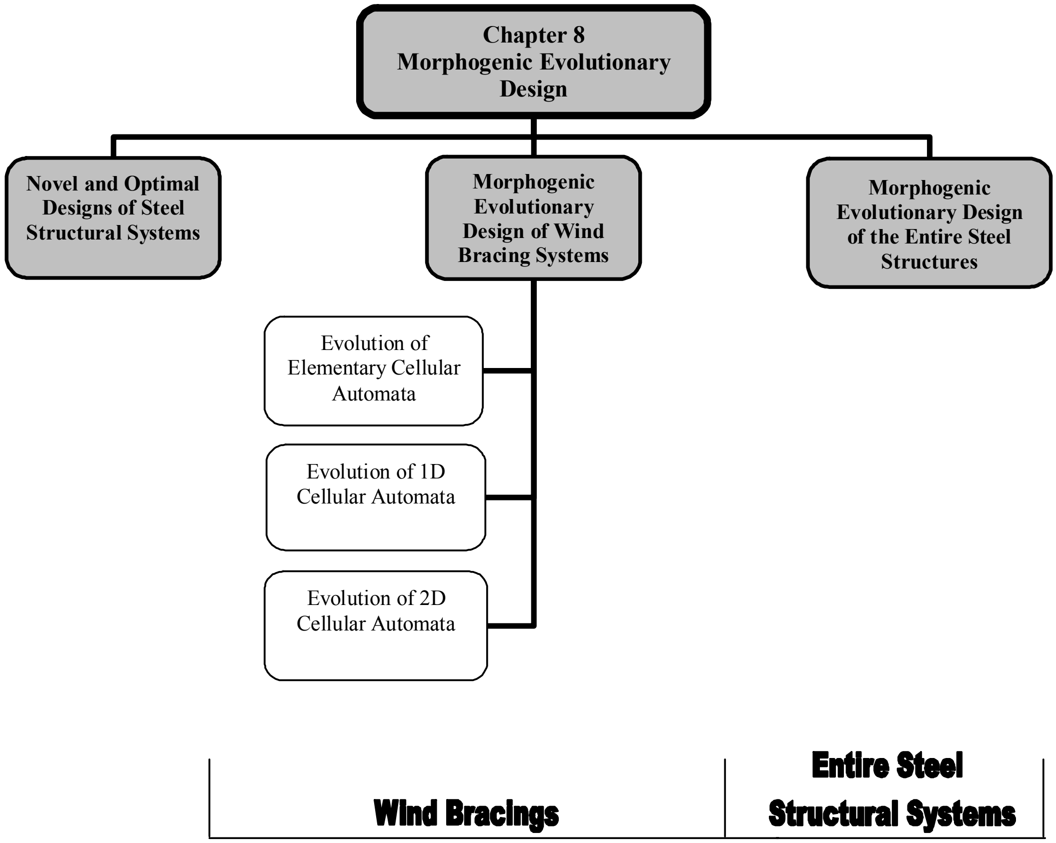


Figure 105. Organization of chapter 8

8.1. Novel and Optimal Designs of Steel Structural Systems

As stated earlier, in this chapter, I describe results of the third and last stage of the Empirical Performance Validation process in which I empirically test the usefulness of the integrated components of EED for producing novel design concepts of steel structural systems and for efficiently optimizing them. By measuring the performance of the integrated components for the example problems, I will test the fundamental research hypothesis of this dissertation (see section 3.3).

Similarly as I did in chapters 6 and 7, I will now refine the fundamental research hypothesis in the context of the design problems considered in this dissertation. In order to do that, I will use the same criteria as in chapter 6 to determine novelty of generated design concepts and the same performance criteria as in chapter 7 to test their optimality. I will also use the same performance measure as in chapters 6 and 7, i.e. the total weight of the structural system, as the objective with respect to which the design concepts will be optimized.

Thus, I can refine the fundamental research question and the fundamental research hypothesis in the specific context of conceptual design of steel structural systems in tall buildings in the following way:

Fundamental Research Question (Refined):

How can one construct an effective method for conceptual design of steel structural systems in tall buildings that would support development of novel designs and their efficient optimization?

Fundamental Research Hypothesis (Refined):

Emergent Engineering Design, a design method in which all major elements of engineering design (i.e. design representation and actual design process) are modeled as complex systems, can effectively produce novel design concepts exhibiting interesting structural shaping patterns and efficiently optimize them with respect to a given objective(s).

This refined fundamental research hypothesis can now be tested empirically. The morphogenic evolutionary design experiments reported in this chapter were conducted to test this hypothesis. In these experiments, generative representations of steel structural systems in tall buildings were evolved by evolutionary algorithms. I have also empirically investigated the impact of several representation specific parameters and evolutionary computation parameters on the quality of produced design concepts.

Table 85 presents the layout of design experiments reported in this chapter. All sections in this chapter are organized to follow this layout. As in chapter 7, the experiments were divided into two major groups depending on the termination criteria used in individual evolutionary optimization runs: short-term experiments and long-term experiments. Extensive sensitivity analyses involving both evolutionary computation parameters and generative representation parameters were conducted during the short-term experiments. The following evolutionary computation parameters were considered: mutation rates, crossover rates, sizes of parent and offspring populations, and the type of the generational model. Only one type of evolutionary algorithm was used in all morphogenic evolutionary design experiments, namely ES, because the experiments reported in chapter 7 have shown that ES significantly outperformed GAs in this problem domain.

The generative representation parameters investigated in the short-term experiments included the type of CA rules (standard vs. totalistic), the radius of the local neighborhood, and the shape of the local neighborhood (2D CAs only). These parameters were identified in chapter 6 as having the biggest impact on the quality of the generated design concepts.

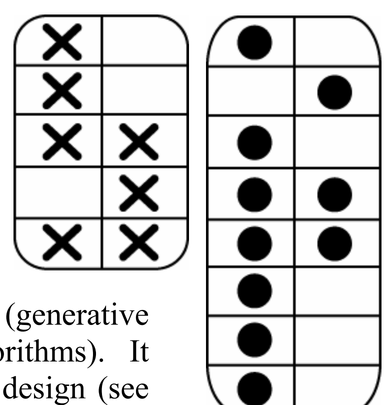
Optimal settings for both evolutionary computation and generative representation parameters were sought in the short-term experiments and, once found, later utilized in the long-term experiments. The performance analysis of morphogenic evolutionary design processes was conducted for both the short-term and the long-term experiments. It included the four performance criteria presented in the bottom part of Table 85.

In general, I will use the same parameters and their values as in chapters 6 and chapter 7 to categorize morphogenic evolutionary design experiments reported in this chapter. Also, two small icons, which were previously introduced in chapters 6 and 7, will be placed at the beginning of each section to indicate the values of the experimental parameters (defined in Table 4 and Table 59) used in the experiments reported in that section.

Table 85. Overview of morphogenic evolutionary design experiments reported in this chapter

		Short-term Experiments	Long-term Experiments
Sensitivity Analysis	Evolutionary computation	Mutation rates	
		Crossover rates	
		Size of parent population	
		Size of offspring population	
		Generational model	
	Generative representations	CA rule type	
		Radius of the local neighborhood	
		Shape of the local neighborhood (2D CA only)	
Performance Analysis	Performance comparison of best design concepts generated in morphogenic evolutionary design experiments and best designs produced in evolutionary optimization processes (chapter 7)		
	Performance comparison of best design concepts generated in morphogenic evolutionary design experiments and best designs produced by generative representations (chapter 6)		
	Performance improvement of the best design concept at the end of an morphogenic evolutionary design process compared to the best design from an initial population		
	Performance improvement of an average design concept at the end of an morphogenic evolutionary design process compared to an average design from an initial population		

8.2. Morphogenic Evolutionary Design of Wind Bracing Systems



In this section, I describe results of the design experiments involving various types of generative representations of wind bracing systems in tall buildings. These representations were evolved by evolutionary algorithms in order to find optimal design concepts. All types of representations considered in this section were introduced earlier in chapter 4.

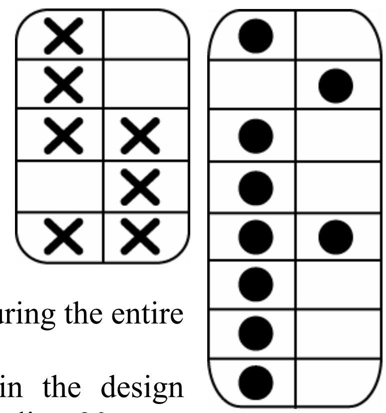
In the design experiments reported in this section, I experimentally investigated the new engineering design paradigm inspired by the developmental processes occurring in nature (generative representations) and the processes of evolution (evolutionary algorithms). It was defined in section 4.3 and named morphogenic evolutionary design (see Definition 4). The obtained results were subsequently compared to the results produced by the parameterized representations of engineering systems (see section 4.2) evolved by evolutionary algorithms (see chapter 7) which constitute the state-of-the-art in engineering design. I also compared the best design concepts produced in the morphogenic evolutionary design experiments to the best design concepts generated in the experiments reported in chapter 6.

The following three subsections describe the results of the morphogenic evolutionary design experiments in which the following types of generative representations were used:

- elementary cellular automata with 2 types of wind bracings elements (as in section 6.2),
- one-dimensional cellular automata with 7 types of wind bracing elements (as in section 6.3), and
- two-dimensional cellular automata (as in section 6.4).

In all three cases, the generative representations were evolved using evolutionary algorithms.

8.2.1. Evolution of Elementary Cellular Automata



The experiments reported in this subsection involved elementary CAs with 2 possible cell values representing 2 types of wind bracing elements. As before (see sections 6.2 and 7.2.1), two groups of wind bracing elements were considered, each consisting of two types of wind bracings. The group No.1 included simple X bracings and no bracings (empty cells) while the group No.2 contained K bracings and no bracings. The remaining members of the steel structural systems in tall buildings, i.e. beams, columns, and supports, were kept the same during the entire morphogenic evolutionary design processes.

The problem parameters and their values that were used in the design experiments described in this section are given in Table 86. As earlier, 30-story buildings with 5 bays were the subject of design. The geometry of the steel structural systems, i.e. heights of the stories and bay widths, were also the same as in the experiments reported in the previous chapters.

The generative representations considered in this subsection consisted of a single 1D design embryo and a single design rule based on an elementary CA rule. A detailed description of this type of generative representation was presented earlier in section 4.4.1.

The experimental results with design concept generators reported in chapter 6 provided some guidance as far as the choice of the most appropriate parameters' settings is concerned. Based on the previous research findings, both elements of the generative representation, i.e. the design embryo and the design rule, were evolved using evolutionary algorithms. Also, only periodic boundary conditions and one location of the design embryo (at the bottom of the structural system) were investigated. Table 87 shows all generative representation parameters and their values which were used in the morphogenic evolutionary design experiments with elementary CAs.

The generative representations based on elementary CAs were evolved by evolutionary algorithms. Similarly as in chapter 7, both short-term and long-term morphogenic evolutionary design processes were conducted. In the short-term processes, the experiments were terminated after 1,000 fitness evaluations. The long-term morphogenic evolutionary design experiments involved as many as 10,000 fitness evaluations per run.

Table 86. Problem parameters and their values used in the morphogenic evolutionary design experiments with elementary CAs

Problem Parameter	*Value(s)*
Problem type	Design of a wind bracing system in a tall building
Number of stories	30
Number of bays	5
Bay width	20 feet (6.01 m)
Story height	14 feet (4.27 m)
Distance between transverse systems	20 feet (6.01 m)
Types of bracing elements	No and Simple X, or No and K
Types of beam elements	Fixed-Fixed
Types of column elements	Fixed-Fixed
Types of supports	Fixed

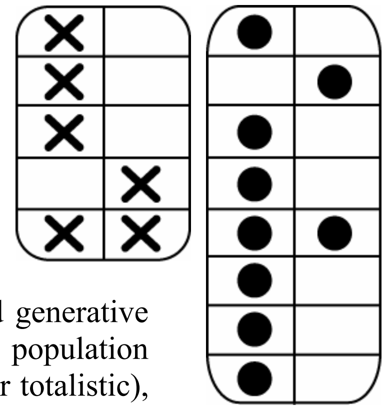
Table 87. Generative representation parameters and their values used in the morphogenic evolutionary design experiments with elementary CAs

Representation Parameter	*Value(s)*
Representation type	Cellular automata
CA dimension	1D
Number of cell states	2
CA rule type	Standard CA rule, or totalistic CA rule
Neighborhood radius	1, or 2
Boundary conditions	Periodic
Design embryo location	Bottom
Design embryo initialization	Random

An extensive parameter search was conducted during the short-term processes. It involved not only evolutionary computation parameters (i.e. parent and offspring population sizes, mutation and crossover rates, and the type of the generational model) but also the generative representations parameters (the type of the CA rule and the radius of the local neighborhood). As in chapter 7, the optimal values of these paramerts were identified and later used in the long-term morphogenic evolutionary design processes.

The results of the short-term and the long-term processes are described in the following subsections.

Short-term Morphogenic Evolutionary Design



In this group of experiments, short-term morphogenic evolutionary design of wind bracing systems in tall buildings was investigated. Table 88 presents evolutionary computation parameters used in the design experiments reported in this subsection. It shows that two kinds of ES were employed: ES(μ+λ) with the overlapping generational model and ES(μ,λ) with the nonoverlapping generational model.

The sensitivity analysis conducted during the short-term experiments involved the following evolutionary computation and generative representation parameters and their values: parent and offspring population sizes, mutation and crossover rates, types of CA rules (standard or totalistic), and the length of the radius of the local neighborhood (1 or 2). The morphogenic evolutionary design processes were repeated 5 times for all combination of parameter values, each time using a different value of the random seed.

The fitness of each design concept was determined, as before, by the total weight of the structural system calculated using the first-order structural analysis. Whenever an infeasible concept was generated, it was assigned a fitness value of 0 (death penalty).

Table 88. Evolutionary computation parameters and their values used in the short-term morphogenic evolutionary design experiments with elementary CAs

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies (ES)
Generational model	Overlapping for ES(μ+λ), Nonoverlapping for ES(μ,λ)
Population sizes (parent, offspring)	(1,5), (5,25), or (50,250) for ES(μ+λ) (5,25) for ES(μ,λ)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.025, 0.1, or 0.3
Crossover (type, rate)	(uniform, 0), (uniform, 0.2), (uniform, 0.5)
Fitness	Total weight of the steel structure (determined by the first-order structural analysis)
Initialization method	Random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	1,000 fitness evaluations
Number of runs	5 in each experiment

The initial population of parents was initialized randomly in all experiments reported here. Each individual was represented as a fixed-length genome. The structure of the genome was introduced earlier in section 4.4.1 (see Figure 28). It consisted of two concatenated parts: the encoding of the design embryo and the encoding of the design rule based on an elementary CA rule. Thus, the genome did not encode a complete design concept, as it was the case with the parameterized representations studied in chapter 7, but instructions (encoded in the design rule) on how to develop a complete design concept from the initial seed (encoded in the design embryo).

In order to evaluate fitness of a design concept encoded using this type of generative representation, a complete configuration of a wind bracing system had to be first fully developed from the design embryo by the application of the design rule. A detailed description of the developmental process was presented in section 4.4.1 (see also Figure 26).

The length of the genome depended on the type of the CA rule (standard vs. totalistic) and the radius of the local neighborhood. In the experiments reported in this section, the genomes had the following lengths: 13 genes and 37 genes for standard CA rules and 9 genes and 11 genes for totalistic CA rules with the radius equal to 1 and 2, respectively. The obtained results are presented in the following subsections.

Optimal Mutation Rates

The initial experiments investigated evolutionary computation parameters in order to determine their optimal values for morphogenic evolutionary design processes. First, an extensive parameter search was conducted to identify the optimal rates of mutation and crossover operators. Table 88 shows that 9 combinations of mutation and crossover rates were considered.

Figure 106 shows typical results regarding the impact of various mutation rates on the fitness of generated design concepts in the short-term experiments with the group No.1 of wind bracing elements (simple X bracings and no bracings). The results presented in this figure were produced in the experiments with ES(5+25) and the generative representation based on standard elementary CA rules with the neighborhood radius equal to 1. The rate of the uniform crossover operator equaled 0.2. The vertical lines represent 95% confidence intervals calculated using Johnson's modified t test.

Figure 106 shows that higher mutation rates were preferred in the short-term morphogenic evolutionary design experiments. Specifically, the best and comparable results were obtained when mutation rates were equal to 0.1 and 0.3. When the lowest mutation rate was employed, i.e. 0.025, then not only the worst results were obtained but they also showed the highest variance. Similar results were observed in the majority of the short-term morphogenic evolutionary design experiments, as it is illustrated graphically in Figure 107. It shows the average fitness values obtained after 1,000 evaluations in the morphogenic evolutionary design experiments with the group No.1 of wind bracing elements (simple X bracings and no bracings). The results presented in this figure were produced by ES(5+25) with two types of elementary CA rules (standard or totalistic) and two lengths of the radius of the local neighborhood (1 and 2). In all cases, the average end-of-run results produced by 9 combinations of mutation and crossover values are presented and sorted with respect to the crossover rate.

A clear pattern can be identified in Figure 107 regarding the impact of the mutation rate on the fitness of the produced design concepts: the higher the mutation rate the better design concepts produced. This pattern was observed in all experiments with standard CA rules with one exception: when the radius equaled 2 and mutation and crossover rates were equal to 0.1 and 0, respectively. The same pattern was observed in the experiments with totalistic CA rules and the radius of 1. However, the situation was different when the radius was increased to 2 and totalistic CA rules were employed. In this case, all mutation rates produced the same results. An explanation of this fact will be presented later at the end of this section.

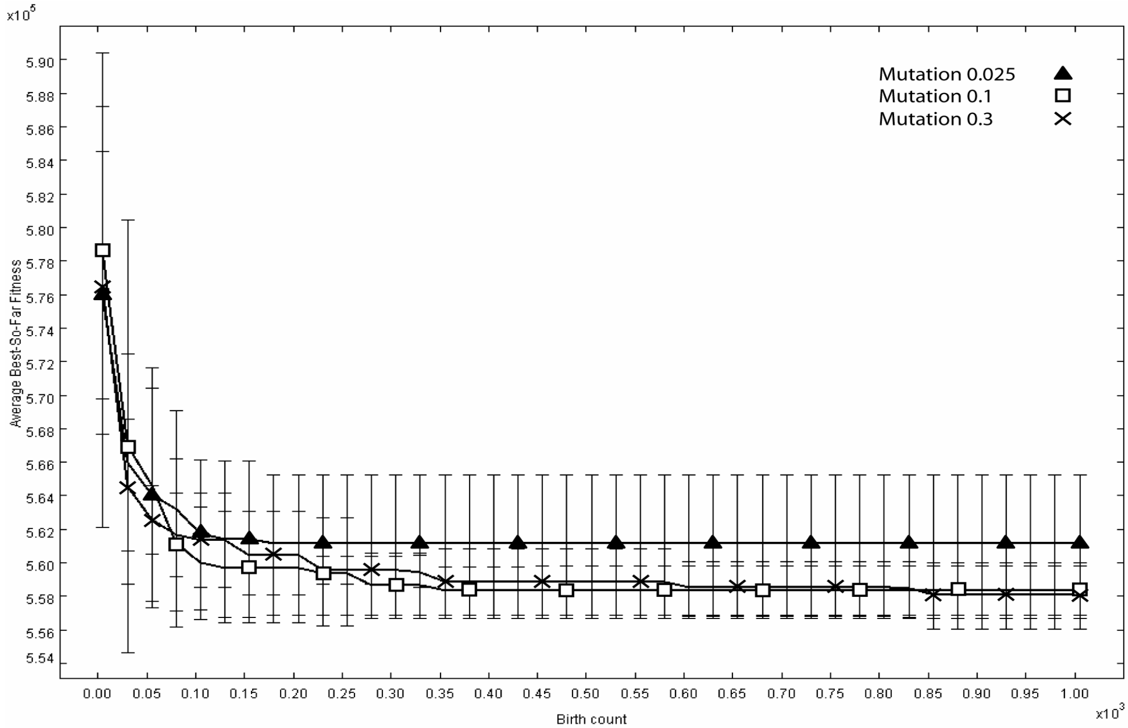


Figure 106. The influence of the mutation rate on the fitness of design concepts generated in the short-term morphogenic evolutionary design experiments with elementary CAs (group No.1 of wind bracing elements)

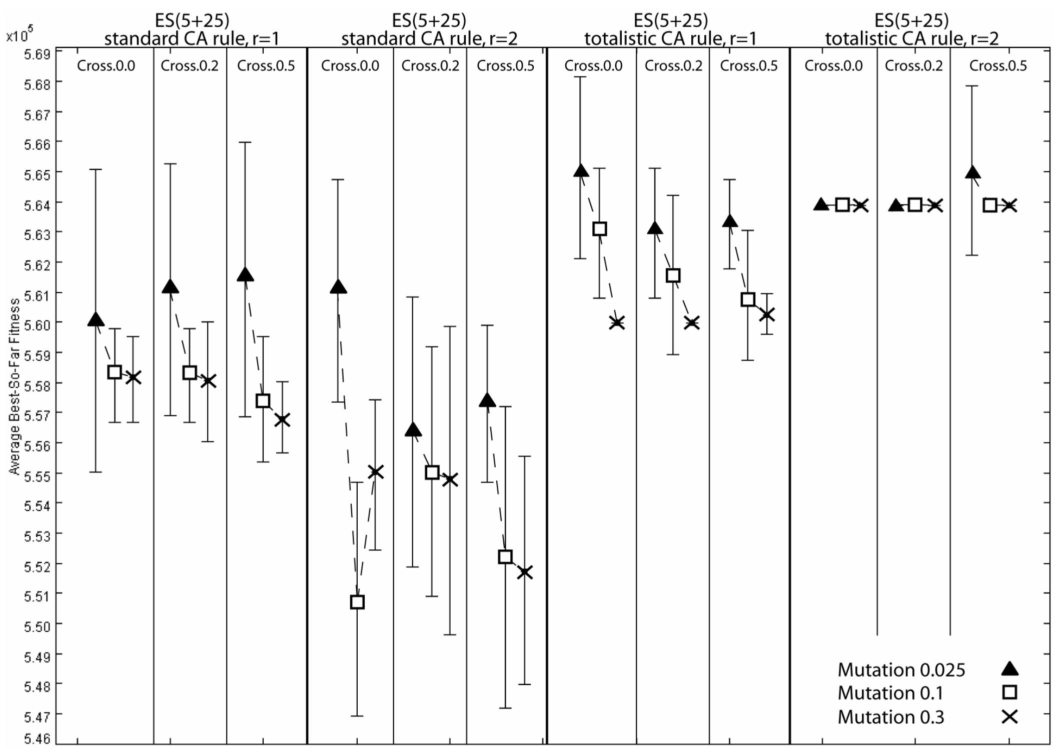


Figure 107. Comparison of the average fitness values (and 95% confidence intervals) obtained after 1,000 fitness evaluations in the short-term morphogenic evolutionary design experiments with elementary CAs

Optimal Crossover Rates

No such pattern was observed for crossover rates. In some cases the best results were obtained when high rates of crossover operator were applied and sometimes when the crossover operator was not used at all. The graph showing these results was, however, omitted.

Optimal Population Sizes

In another group of experiments, the impact of the sizes of parent and offspring populations on the fitness of generated design concepts was tested. Figure 108 shows the results of the experiments in which 3 combinations of parent and offspring population sizes were used: ES(1+5), ES(5+25), and ES(50+250). All other evolutionary computation parameters and generative representation parameters had the same values in these experiments. Specifically, the mutation and crossover rates were equal to 0.3 and 0.2, respectively, and standard CA rules with the radius of 1 were employed.

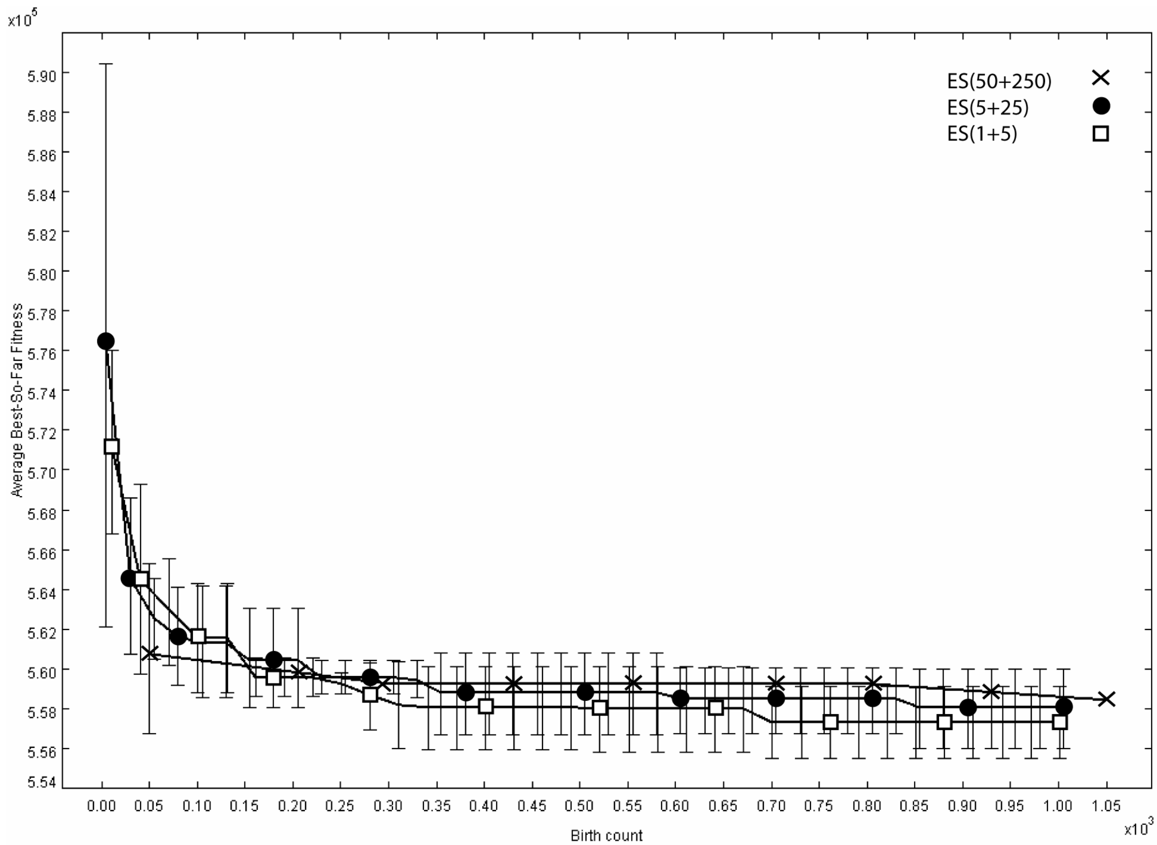


Figure 108. The influence of the sizes of parents and offspring populations on the progress of the short-term morphogenic evolutionary design processes with the group No.1 of wind bracing elements

Figure 108 shows that all combinations of the population sizes produced comparable results in terms of the average best-so-far fitness. The substantial differences occurred only in the level of variance. Small population sizes, i.e. ES(1+5) and ES(5+25) showed higher variance than the large population sizes ES(50+250).

Optimal Generational Model

Finally, Figure 109 compares the results of experiments in which the impact of the type of the generational model on the fitness of produced design concepts was investigated. Here, two kinds of ES were employed: ES(5,25) with the nonoverlapping generation model and ES(5+25) with the overlapping generational model. As earlier, all other parameters's values were kept the same.

Figure 109 clearly shows that ES with the overlapping and nonoverlapping generational models produced almost identical results in terms of both the average best-so-far fitness and the variance of produced results.

Optimal Generative Representation Parameters

The impact of the generative representation parameters, i.e. the type of the CA rule and the length of the radius of the local neighborhood, was dramatically different for each group of wind bracing elements. Hence, the obtained results are discussed separately for the group No.1 (simple X bracings) and the group No.2 (K bracings).

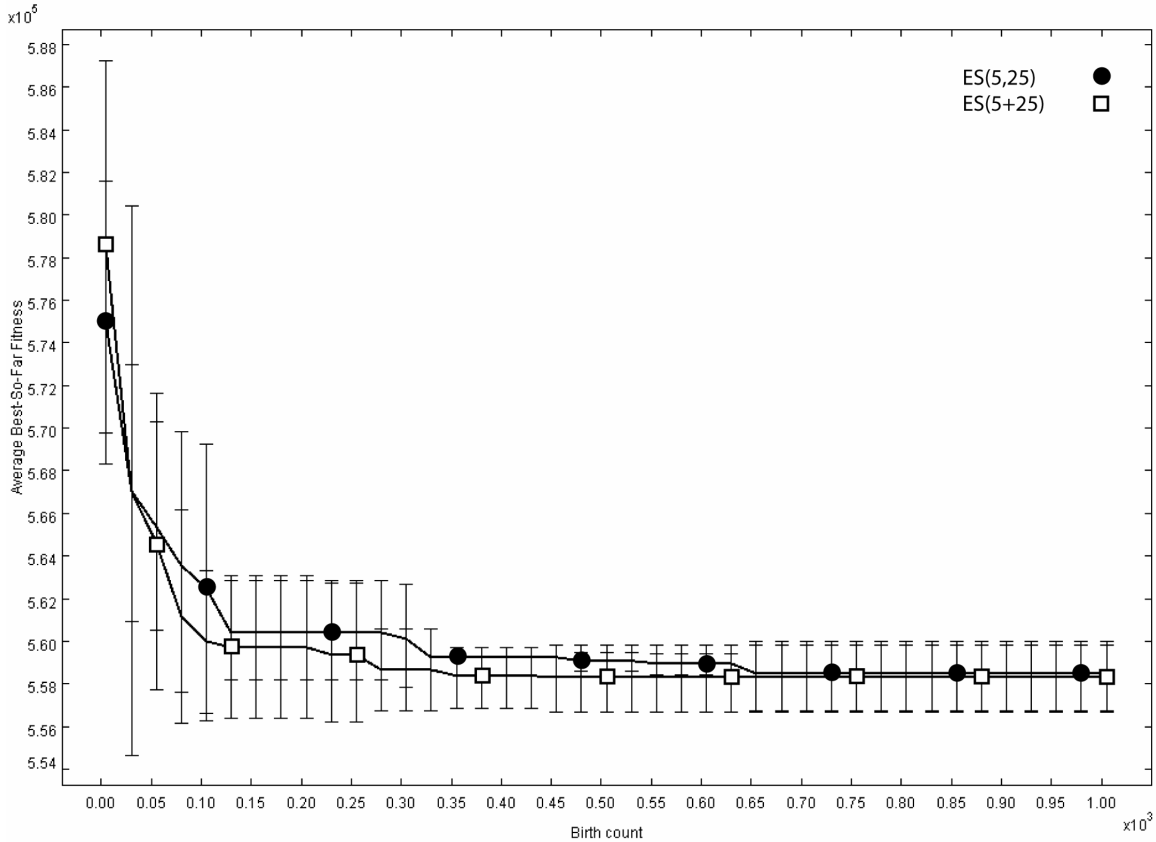


Figure 109. The influence of the type of the generational model on the short-term morphogenic evolutionary design processes with the group No.1 of wind bracing elements

Simple X Bracings

Figure 110 shows typical results regarding the impact of both generative representation parameters on the average best-so-far fitness obtained in the design experiments with the first group of wind bracing elements. Specifically, these results were produced by ES(5+25) with the mutation rate equal to 0.1 and crossover rate equal to 0.2. Vertical lines denote 95% confidence intervals.

Figure 110 clearly shows that standard CA rules produced better results than totalistic CA rules in the morphogenic evolutionary design experiments with the first group of wind bracing elements. It also revealed that the impact of the increased length of the radius of the local neighborhood on the fitness of design concepts was distinct for each type of elementary CA rules. Standard CA rules with the radius equal to 2 produced significantly better results than the ones with the radius equal to 1. The situation was, however, different for totalistic CA rules. In this case, the design experiments with the smaller radius produced slightly better results compared to the results generated with the radius equal to 2.

Interestingly, there were also significant differences among variances obtained in the design experiments shown in Figure 110. The results with the largest variance were produced by standard CA rules with the radius equal to 2. On the other hand, the smallest variance (in fact, no variance at all because all 5 runs produced exactly the same best design concept) was obtained in the design experiments with totalistic CA rules and the radius equal to 2.

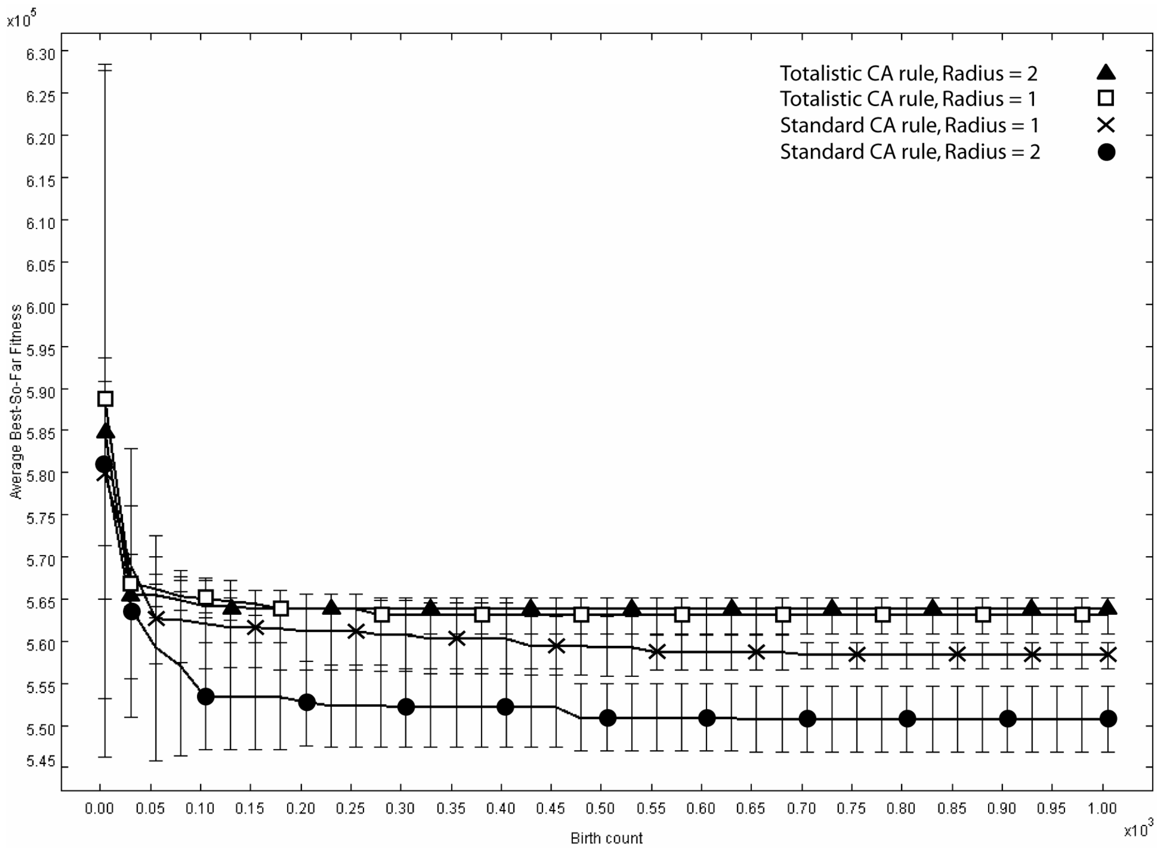


Figure 110. The influence of the type of the CA rule and the radius of the local neighborhood on the fitness of design concepts generated in the short-term morphogenic evolutionary design experiments with elementary CAs (group No.1 of wind bracing elements)

K Bracings

As I mentioned earlier, the results were different when the second group of wind bracing elements (K bracings and no bracings) was employed. Figure 111 shows typical results obtained in these experiments. Here, ES(5+25) was used with the mutation rate equal to 0.1 and the crossover rate equal to 0.2. It clearly shows that both standard and totalistic CA rules produced the design concepts of almost identical fitness. The differences among the design processes utilizing standard and totalistic CA rules occurred only in the initial stages of evolution (up to 100 fitness evaluations). Specifically, totalistic CA rules generated better solutions faster than standard CA rules. The impact of the increased radius of the local neighborhood was also restricted to the initial stages of the design processes only (first 100 fitness evaluations). The longer radius did not influence the average fitness of the best design concepts obtained in the short-term experiments.

Unlike the experiments with the group No.1 of wind bracing elements, the experiments with K bracings did not show any significant differences among variances of the results. The variance of the fitness of generated design concepts was high in the initial stages of evolution (up to 50 fitness evaluations), particularly for totalistic CA rules, but it was quickly reduced and after 300 fitness evaluations became almost negligible.

K Bracings - Performance Improvement

As discussed earlier, the results of the morphogenic evolutionary design processes with elementary CAs substantially differed for the two groups of wind bracing elements. These differences could also be identified when the performance improvements and optimal design concepts produced in the morphogenic evolutionary design experiments were compared to the ones obtained in the evolutionary optimization processes (see chapter 7).

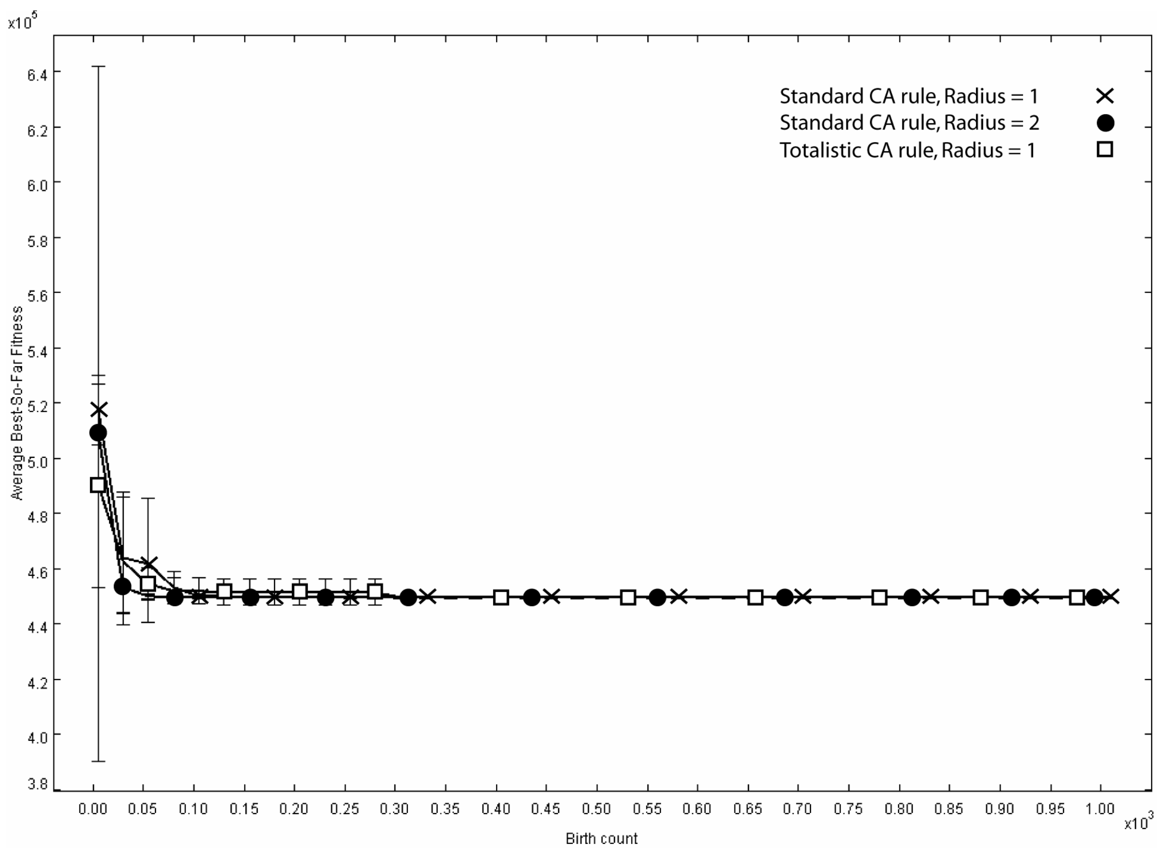


Figure 111. The influence of the type of the CA rule and the radius of the local neighborhood on the fitness of design concepts generated in the short-term morphogenic evolutionary design experiments with elementary CAs (group No.2 of wind bracing elements)

Figure 112 compares the average best-so-far fitness of the design concepts produced in the evolutionary optimization experiments (see section 7.2.1) and the morphogenic evolutionary design experiments with the group No.2 of wind bracing elements (K bracings). In the former case, the parameterized representations of wind bracing systems were used (see section 4.2). In the latter case, the generative representations based on elementary CAs (standard CA rules and totalistic CA rules) were employed. In both cases, the optimal values of evolutionary computation parameters were applied, i.e. ES(5+25) with the mutation rate equal to 0.025 and the crossover rate equal to 0.2 for the parameterized representations and ES(5+25) with the mutation rate equal to 0.1 and the crossover rate equal to 0.2 for the generative representations.

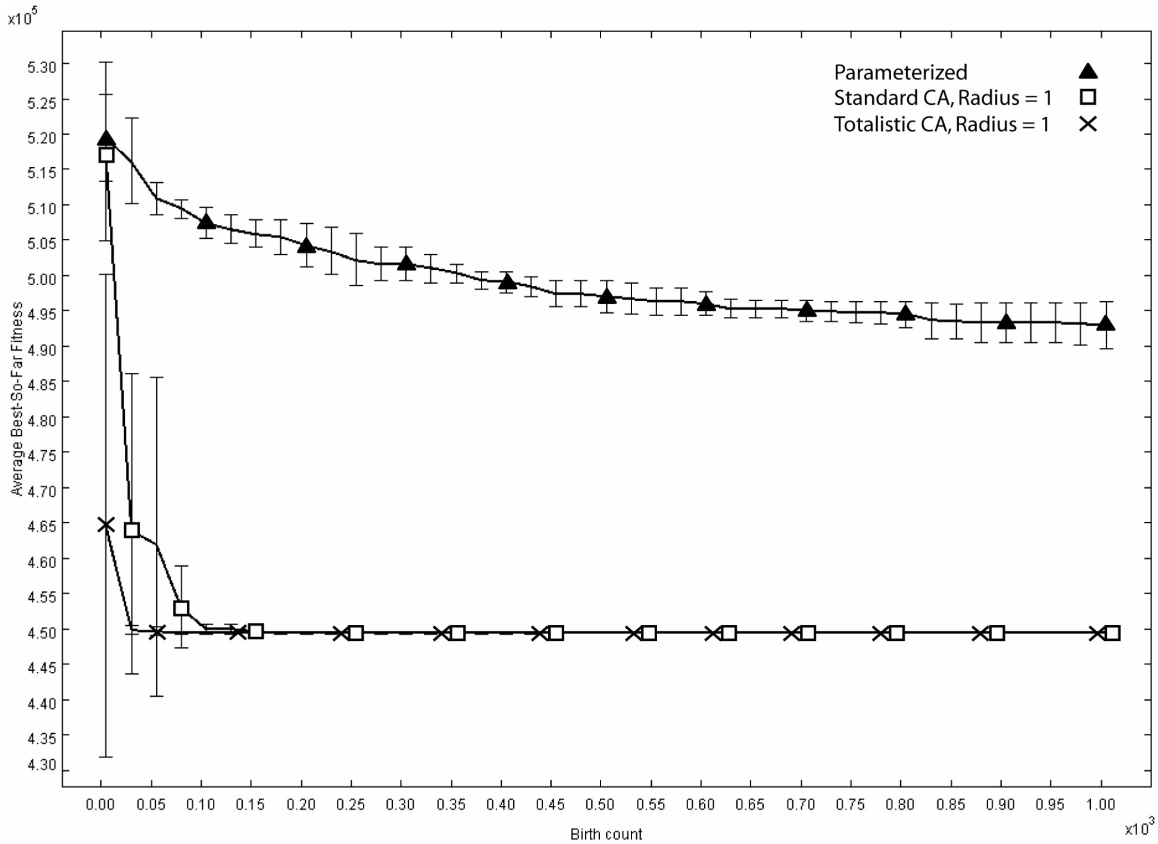


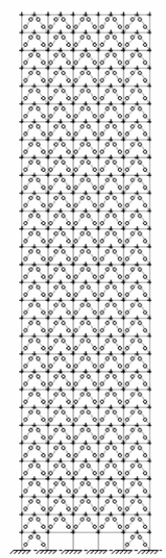
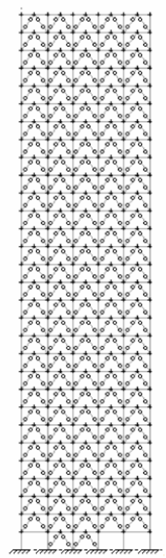
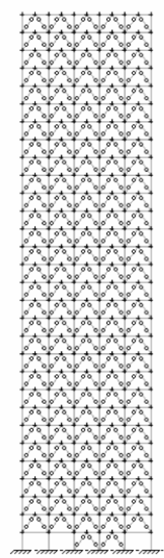
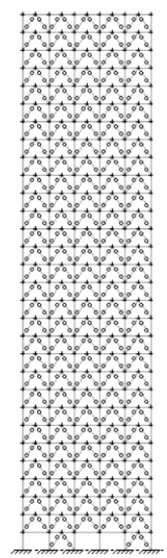
Figure 112. Comparison of the average best-so-far fitness produced in the evolutionary optimization experiments (parameterized representations) and morphogenic evolutionary design experiments with elementary CAs (standard CA rules and totalistic CA rules) for the group No.2 of wind bracing elements

Figure 112 shows that the morphogenic design processes significantly outperformed the evolutionary optimization processes when the group No.2 of wind bracing elements was employed. The performance improvement obtained after 1,000 evaluations beetween the generative representations based on elementary CAs and the parameterized representations exceeded 43,000 lbs., or 8.5 percent. The average performance improvement achieved in the morphogenic evolutionary design experiments was equal to about 68,200 lbs., or 13.2 percent, for standard CA rules and 59,700 lbs., or 11.7 percent, for totalistic CA rules compared to 26,600 lbs., or 5.1 percent, obtained in the evolutionary optimization processes.

K Bracings - Optimal Designs

The best design concepts of wind bracing systems composed of K bracings are presented in Table 89. The fitness of the best design concept found in the morphogenic design experiments was equal to 449,376 lbs. It was about 40,500 lbs., or 8.2 percent, better than the best design produced in the short-term evolutionary optimization experiments (see Table 64). At the same time, it achieved the same fitness as the best design concept found in the exhaustive search of elementary CA rules reported in chapter 6 (see Table 37).

Table 89. Best design concepts of wind bracing systems composed of K bracings generated in the short-term morphogenic evolutionary design experiments with elementary CAs

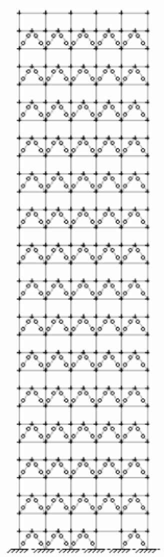
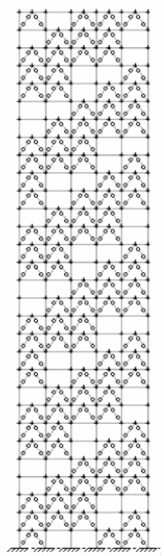
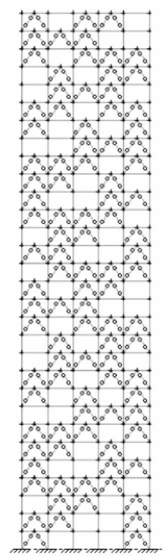
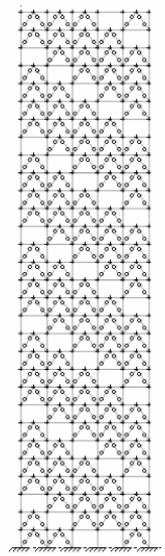
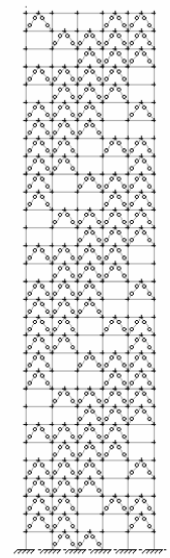
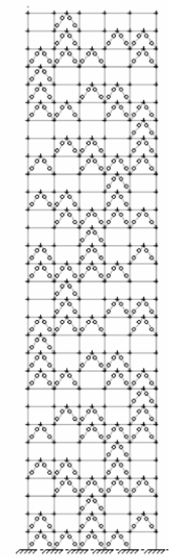
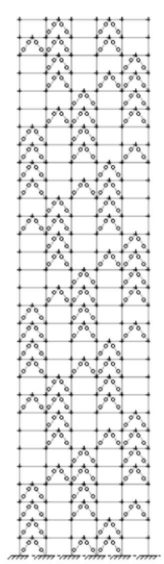
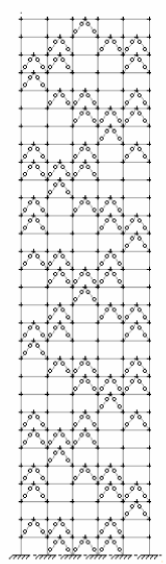
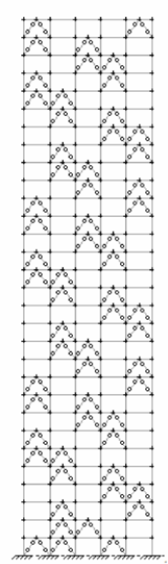
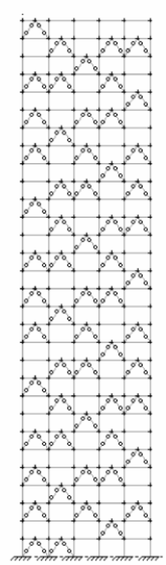
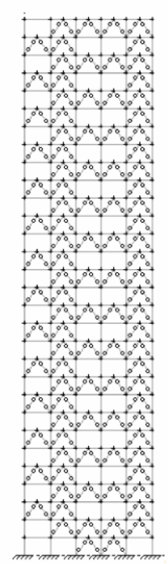
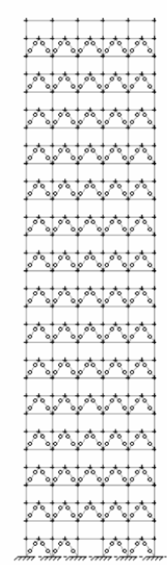
			
449,376 4.9888	449,521 4.9813	449,521 4.9813	449,521 4.9900

K Bracings - Structural Shaping Patterns

Thus, morphogenic evolutionary design processes significantly outperformed evolutionary optimization processes in this problem domain. They also generated interesting structural shaping patterns of good performance that were qualitatively different than the patterns obtained in the optimization experiments reported in chapter 7. Several examples of the design concepts of wind bracing systems with interesting structural shaping patterns composed of K bracings are presented in Table 90.

Table 90 contains several interesting patterns which were previously identified in chapter 6 in the design experiments with elementary CAs as well as a few novel ones. For example, the patterns of horizontal trusses (see the 1st and 12th design concepts in Table 90) and of macro bracings (see the 2nd, 4th, and 5th design concepts in Table 90) have been previously found and described in chapter 6. On the other hand, the structural shaping patterns exhibited by the 6th, 7th, 8th, 9th, and 10th design concepts and produced by the standard CA rules with the radius equal to 2 represent novel configurations of K bracings.

Table 90. Interesting structural shaping patterns generated in the short-term morphogenic evolutionary design experiments with elementary CAs and the group No.2 of wind bracing elements

 532,338 7.1131	 514,987 5.1470	 736,955 3.8891	 523,408 4.8811	 521,790 5.0936	 521,518 5.8954
 517,393 5.7572	 719,535 4.3892	 522,037 6.4854	 529,192 6.4219	 523,767 5.6315	 532,338 7.1178

Simple X Bracings - Performance Improvement

As discussed earlier, the results of the morphogenic evolutionary design processes with the group No.1 of wind bracing elements (simple X bracings and no bracings) produced somewhat different results, particularly with respect to the optimization of wind bracing systems. Figure 113 compares the average best-so-far fitness of the design concepts produced in morphogenic evolutionary design experiments with elementary CAs (standard CA rules with the radii of the local neighborhood equal to 1 or 2) to the results obtained in evolutionary optimization experiments reported in chapter 7. In both cases, the optimal values of evolutionary computations parameters were employed.

Figure 113 clearly shows that even though the morphogenic evolutionary design processes outperformed the evolutionary optimization processes in the initial stages of evolution (up to 200 evaluations), they later produced inferior results. The average performance improvement after 1,000 fitness evaluations was about 14,700 lbs., or 2.7 percent, worse (i.e. negative) for standard CA rules with the radius equal to 1 and about 8,600 lbs., or 1.6 percent, worse when the radius was equal to 2 compared to the average fitness obtained in the evolutionary optimization experiments.

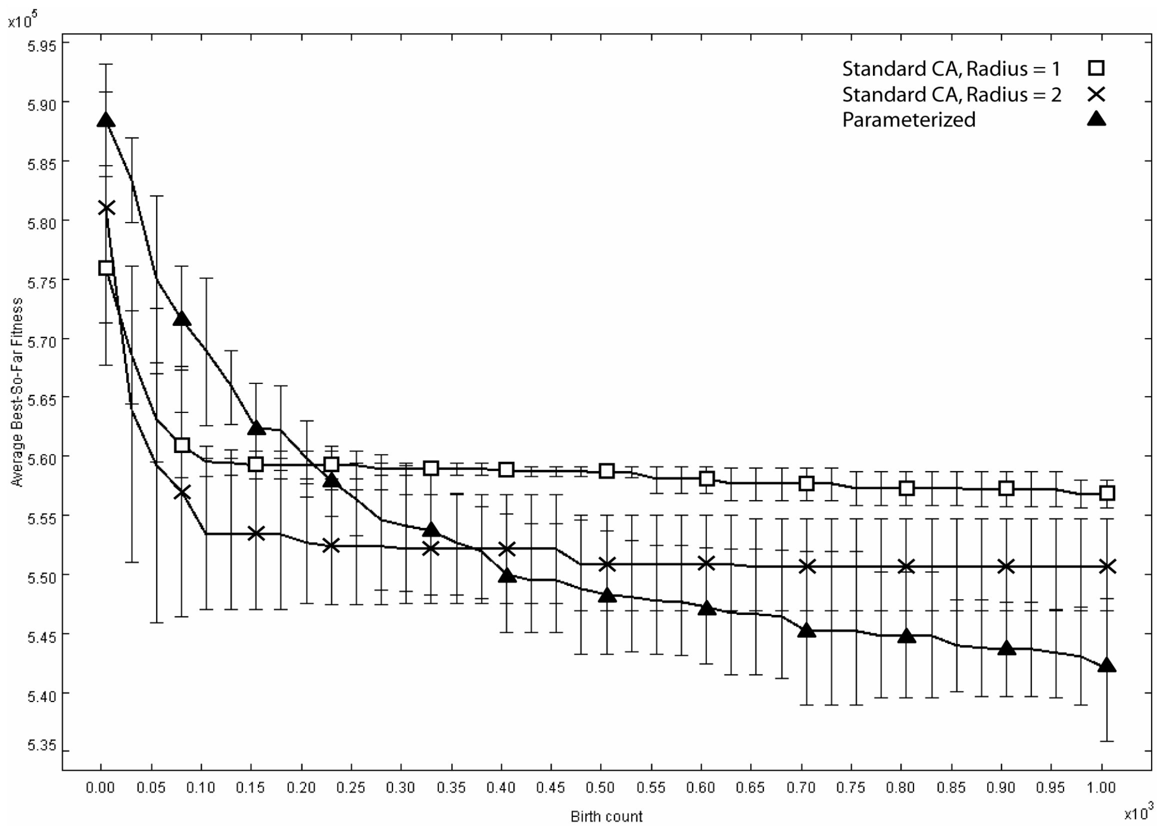
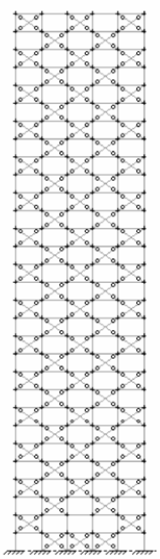
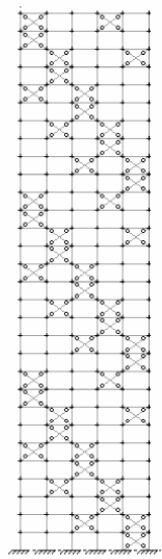
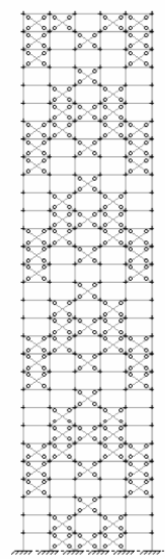
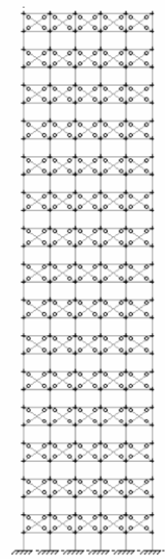


Figure 113. Comparison of the average best-so-far fitness produced in the evolutionary optimization experiments (parameterized representations) and morphogenic evolutionary design experiments with elementary CAs (standard CA rules and totalistic CA rules) for the group No.1 of wind bracing elements

Simple X Bracings – Optimal Designs

The best design concepts generated in the morphogenic design experiments with the group No.2 of wind bracing elements are presented in Table 91. It shows four best design concepts produced in the experiments with four combinations of the generative representation parameters. The overall best design concept produced in the short-term morphogenic design experiments was generated by standard CA rules with the radius of the local neighborhood equal to 2. Its fitness was 548,243 lbs. and it was more than 16,000 lbs., or 3 percent, worse than the best design concept produced in the short-term evolutionary optimization experiments (see Table 62 in chapter 7). At the same time, it slightly outperformed (by about 2,000 lbs., or 0.4 percent) the best design concept generated by elementary CAs (see Table 31 in chapter 6).

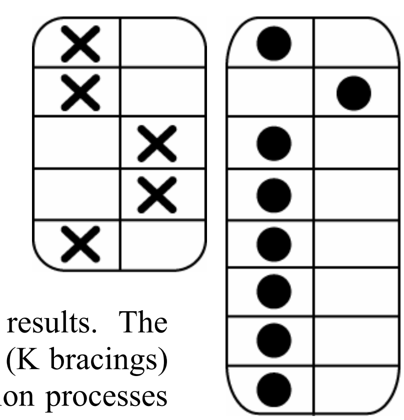
Table 91. Best design concepts of wind bracing systems composed of X bracings generated in the short-term morphogenic evolutionary design experiments with elementary CAs

Standard CA Radius = 1	Standard CA Radius = 2	Totalistic CA Radius = 1	Totalistic CA Radius = 2
			
556,177 4.4727	548,283 5.4386	559,982 4.8465	563,865 6.6963

The remaining 3 design concepts had worse fitness than the best concepts generated in the design experiments with elementary CAs and the best design concepts produced in the evolutionary optimization experiments. On the other hand, they exhibited interesting and diverse structural shaping patterns, including the checkerboard pattern, the horizontal truss pattern, a the pattern identical to the one which was generated by the elementary rule 105 (see Table 35).

In the next subsection, I will investigate the long-term morphogenic evolutionary design processes and test whether they can produce better design concepts than the long-term evolutionary optimization processes described in chapter 7. I will also compare the results of the long-term and the short-term morphogenic evolutionary design experiments.

Long-term Morphogenic Evolutionary Design



In this subsection, the results of the long-term morphogenic evolutionary design processes involving elementary CAs are described. As before (see chapter 7), the length of the long-term processes was siginifcantly larger than of the short-term processes and involved 10,000 fitness evaluations. The obtained results are presented below.

In the previous subsection, I reported that the short-term morphogenic evolutionary design experiments with the two groups of wind bracing elements produced dramatically different results. The short-term morphogenic design experiments with the group No.2 (K bracings) significantly outperformed the short-term evolutionary optimization processes by producing on average 8.5 percent fitter design concepts. Moreover, the optimal design concepts were found very quickly, i.e. within the first 100 fitness evaluations when totalistic CA rules were used.

Performance Improvement

Figure 114 compares the results of the short-term morphogenic evolutionary design experiment with totalistic CA rules to the results produced in the long-term evolutionary optimization using parameterized representations (see section 7.2.1). It is clear that even the long-term evolutionary optimization processes were significantly inferior to the short-term morphogenic evolutionary design processes in this problem domain. The average fitness achieved in the long-term evolutionary optimization experiment was more than 35,000 lbs., or 7.8 percent, worse than the average fitness produced in the short term morphogenic evolutionary design experiment. Thus, the morphogenic evolutionary design processes not only produced significantly better results than the evolutionary optimization processes but they also achieved this performance in a fraction of a computational effort required by the latter.

The short-term morphogenic evolutionary design experiments with the group No.1 of wind bracing elements (simple X bracings) showed that the obtained results were worse than the ones obtained in the short-term evolutionary optimization experiments. In the experiments described below I investigated if and by how much the performance of the morphogenic evolutionary design processes can be improved in the long-term processes.

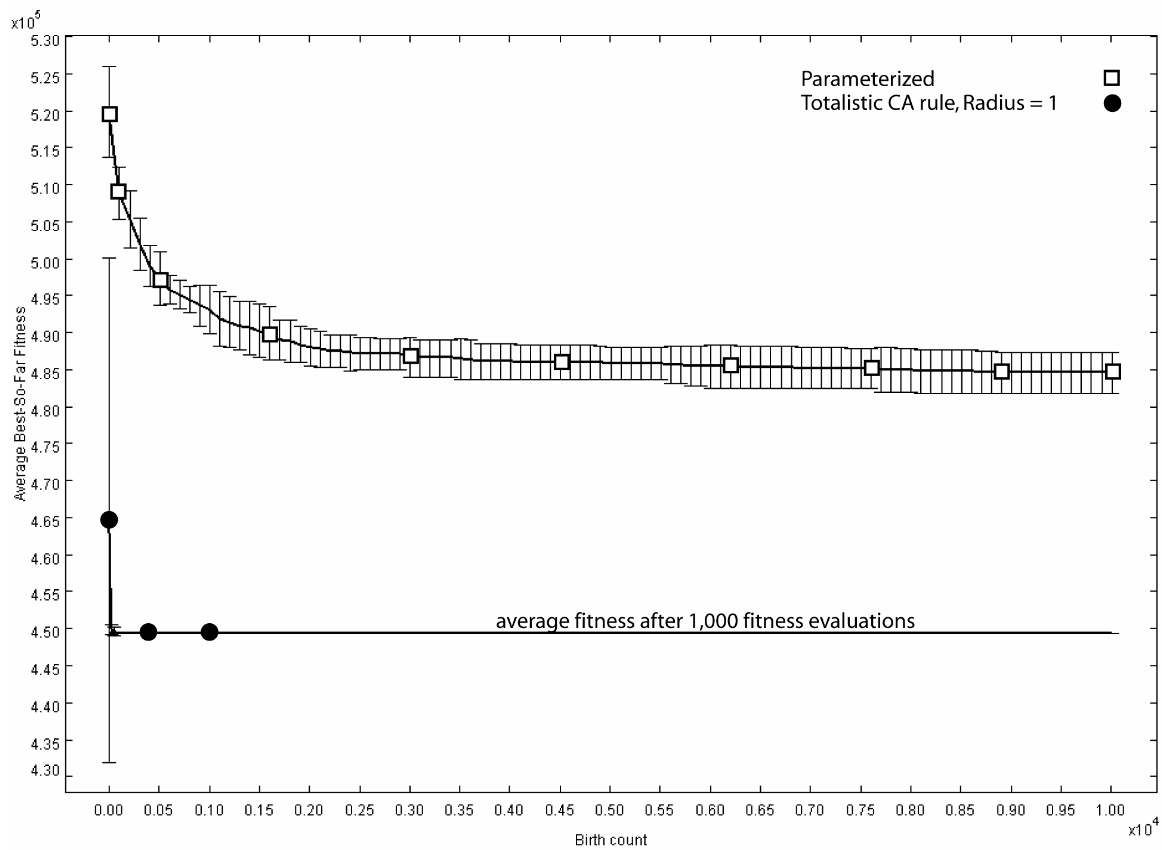


Figure 114. Comparison of the average best-so-far fitness produced in the long-term evolutionary optimization experiments (parameterized representations) and the short-term morphogenic evolutionary design experiments with elementary CAs (totalistic CA rules) with the group No.2 of wind bracing elements (K bracings)

The optimal values of evolutionary computation and the generative representation parameters, identified in the short-term experiments, were employed in the long-term processes with the group No.1 of wind bracing elements. The evolutionary computation parameters and their values are presented in Table 92 while the generative representation parameters are shown in Table 93.

Table 92. Evolutionary computation parameters and their values used in the long-term morphogenic evolutionary design experiments with elementary CAs

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies (ES)
Generational model	Overlapping for ES(μ+λ)
Population sizes (parent, offspring)	(5,25)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.1
Crossover (type, rate)	(uniform, 0.2)
Fitness	Total weight of the steel structure (determined by the first-order structural analysis)
Initialization method	Random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	10,000 fitness evaluations
Number of runs	5 in each experiment

Table 93. Generative representation parameters and their values used in the long-term morphogenic evolutionary design experiments with elementary CAs

Representation Parameter	*Value(s)*
Representation type	Cellular automata
CA dimension	1D
Number of cell states	2
CA rule type	Standard CA rule
Neighborhood radius	2
Boundary conditions	Periodic
Design embryo location	Bottom
Design embryo initialization	Random

Figure 115 compares the results of the long-term morphogenic evolutionary design experiment to the long-term evolutionary optimization experiment with the group No.1 of wind bracing elements. It clearly shows that, as in the short-term processes, the latter significantly outperformed the former in the long-term experiments. The difference between the average best-so-far fitness obtained after 10,000 evaluations exceeded 21,500 lbs., or 4.1 percent. Also, the average performance improvement between the long-term and the short-term morphogenic design experiments was very small and equal to about 2,300 lbs., or 0.4 percent. It was a significantly smaller improvement than the one obtained in the evolutionary optimization experiments where the corresponding improvement level was equal to 15,500 lbs., or 2.8 percent.

Optimal Designs

The best design concepts generated in the long-term morphogenic evolutionary design experiments are presented in Table 94. The fitness of the overall best design concept was equal to 547,428 lbs. It was more than 27,000 lbs., or 5.2 percent, worse than the best design concept produced in the evolutionary optimization experiments (see Table 62 in chapter 7). However, it was almost 3,000 lbs., or 0.5 percent, better than the best design concept generated by elementary CAs (see Table 31 in chapter 6).

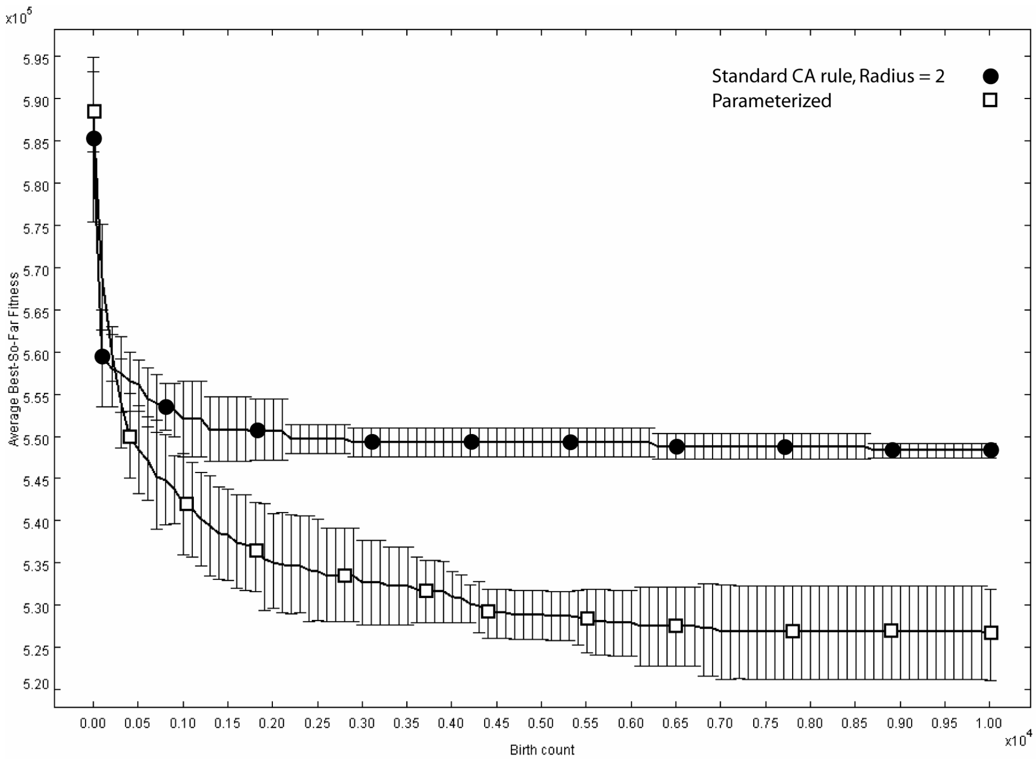
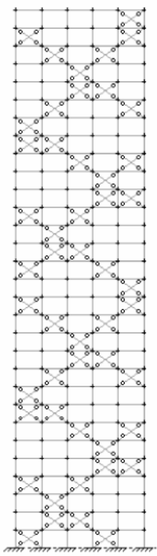
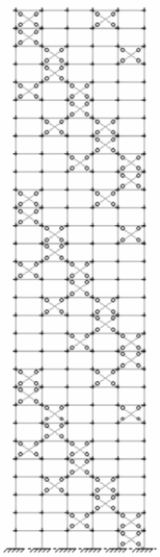
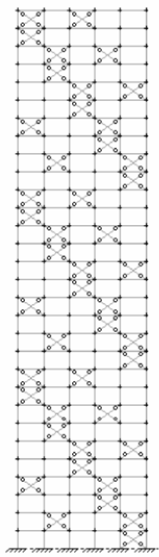


Figure 115. Comparison of the average best-so-far fitness produced in the long-term evolutionary optimization experiments (parameterized representations) and the long-term morphogenic evolutionary design experiments with elementary CAs and the group No.1 of wind bracing elements

Structural Shaping Patterns

Even though the best design concepts in the morphogenic evolutionary design experiments with the group No.1 of wind bracing elements did not represent better solutions in terms of the total weight of the steel structural system, they generated interesting and qualitatively different structural shaping patterns. A prominent example of that is the best design shown in Table 94 in which an emergent pattern of crossed macro bracings can be easily identified. The two other design concepts exhibit interesting variations of the macro bracing pattern which were not found in the design experiments with elementary CAs (see chapter 6).

Table 94. Best design concepts of wind bracing systems composed of X bracings generated in the long-term morphogenic evolutionary design experiments with elementary CAs

 547,428 5.4582	 548,283 5.4386	 549,373 5.5352

Summary

In this section, I described the results of the morphogenic evolutionary design experiments with the simplest generative representations based on elementary CAs. The number of types of wind bracing elements considered in the experiments was limited to 2 only. Similarly as in sections 6.2 and 7.2.1, two groups of wind bracing elements, each consisting of two types of wind bracings, were studied separately in the morphogenic evolutionary design processes. The obtained results were dramatically different for these two design problems.

Initial sensitivity analyses conducted in the short-term experiments have shown that higher mutation rates, i.e. 0.1 or 0.3, were preferred in the morphogenic evolutionary design processes for both design problems. No such pattern was observed for the crossover operator. In some cases the best results were produced when high crossover rates were used and sometimes when crossover was not applied at all. It was also found that neither the size of the parent and offspring populations nor the type of the generational model had any significant impact on the average fitness of the generated design concepts.

The impact of the generative representation parameters, i.e. the type of CA rules and the radius of the local neighborhood, on the fitness of produced design concepts was different for each of the two design problems. For K bracings, both types of CA rules generated the same end results but totalistic CA rules found the optimal solutions much faster (within 100 fitness evaluations) than standard CA rules. For simple X bracings, standard CA rules produced significantly better results than totalistic CA rules. Moreover, there were important differences in the fitness of generated design concepts when the increased length of radius of the local neighborhood was employed. In this case, standard CA rules with the radius of 2 produced the overall best results. At the same time, totalistic CA rules with the radius equal to 2 produced the overall worst results. My hypothetical explanation of these facts is presented below.

The two design problems, i.e. design of a wind bracing system composed of simple X bracings and design of a wind bracing system composed of K bracings, represent two different classes of problems. The optimal solutions for the latter problem have a form of regular configurations of K bracings (variations of the fully-braced pattern). On the contrary, the optimal design concepts for the former problem exhibit very elaborate configurations/shaping patterns of simple X bracings.

The generative representations based on elementary CAs reduce the sizes of the design spaces and thus significantly limit the number of possible configurations of wind bracing elements that can be generated. The amount of the reduction of the design space is affected by two parameters: the type of the CA rule and the radius of the local neighborhood. Standard CA rules offer much bigger potential for developing elaborate patterns but at a cost of significantly larger sizes of CA rule spaces. Thus, they are oriented more towards novelty. On the contrary, totalistic CA rules rapidly decrease the sizes of the CA rules spaces but at a cost of generating only a small fraction of the patterns that can be produced by standard CA rules. Due to the averaging process, totalistic CA rules are biased towards generation of fairly uniform or periodic patterns.

An increased size of the radius of the local neighborhood affects the standard and totalistic rule spaces in two different ways. In the former case, it increases the number of configurations that can reached but at the same time makes the standard CA rule spaces even larger. In the latter case, it increases the amount of averaging and hence further decreases the number of configurations that can be produced. Thus, totalistic CA rules with the longer radius of the local neighborhood are even more biased towards generation of uniform patterns.

Figure 116 shows the average performance improvements obtained in the short- and long-term morphogenic evolutionary design experiments for both groups of wind bracing elements. It clearly illustrates that the morphogenic evolutionary design processes achieved high levels of performance improvement in the initial stages of evolution (the short-term improvements are almost the same as the long-term improvements). Hence, they much faster produced good results than the parameterized representations discussed in the previous chapter. They also achieved siginificantly higher levels of performance improvement in the case of K bracings. Figure 116 shows that in this case the improvement level exceeded 11.5% in the short-term experiments.

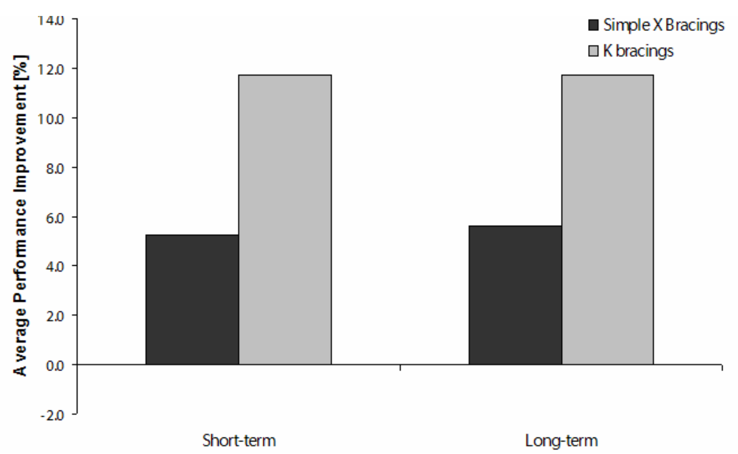


Figure 116. Comparison of the average performance improvements produced in the morphogenic evolutionary design of wind bracing systems with 2 types of bracing elements in the short-term and long-term experiments

The results of the short-term and the long-term morphogenic evolutionary design experiments were also compared to the results obtained in the evolutionary optimization experiments (see chapter 7). Figure 117 shows the averrage performance improvements between the morphogenic evolutionary design and evolutionary optimization achieved in the conducted experiments. It clearly shows that morphogenic evolutionary design processes significantly outperformed the evolutionary optimization processes in the design problem in which K bracings were used. They, however, produced inferior results (shown in this figure as negative values of the performance improvements) when simple X bracings were employed.

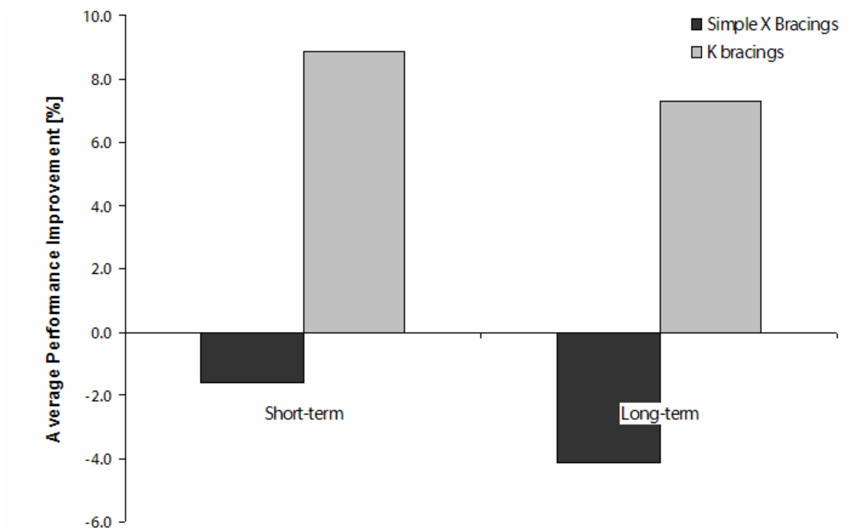


Figure 117. Comparison of the average performance improvements produced in the morphogenic evolutionary design and evolutionary optimization of wind bracing systems with 2 types of bracing elements in the short-term and long-term experiments

Finally, Figure 118 illustrates the performance improvements between the best design concepts of wind bracing systems produced in morphogenic evolutionary design experiments and the best designs generated by elementary CAs (see section 6.2). It shows that the morphogenic evolutionary design experiments produced slightly better (by about 0.5%) designs of wind bracing systems composed of simple X bracings than elementary CAs. On the other hand, both morphogenic evolutionary design processes and elemenatry CAs produced exactly the same best design concepts composed of K bracings (performance improvement was equal to 0 in both the short-term and the long-term experiments).

In this section, I also showed that morphogenic evolutionary design processes generated interesting structural shaping patterns for both design problems. In the case of design concepts composed of simple X bracings, a unique structural shaping pattern of crossed macro bracings has been discovered. This pattern has not been found in the design experiments with elementary CAs reported in chapter 6.

In the next section, I will slightly scale up the design problem considered by the morphogenic evolutionary design processes by using the entire selection of 7 types of wind bracing elements rather than only 2 types, as I did in this section.

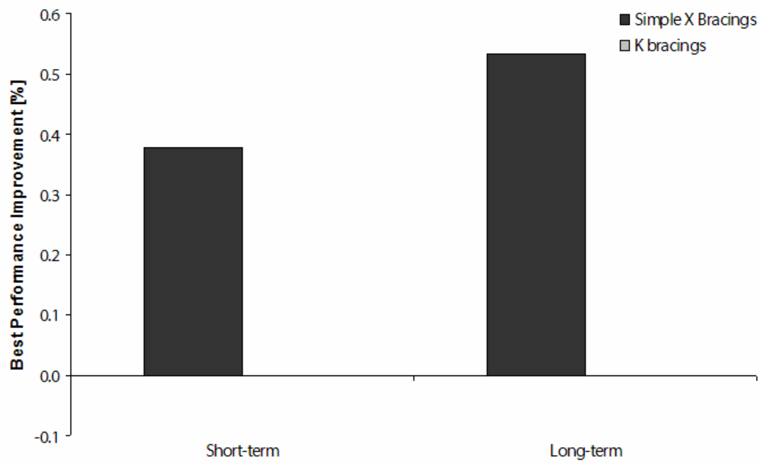
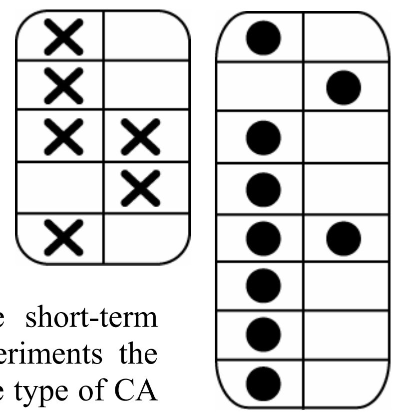


Figure 118. Comparison of the performance improvements between the best designs produced in the morphogenic evolutionary design experiments and the best designs generated by elementary CAs

8.2.2. Evolution of 1D Cellular Automata



In this section, I report the results of the morphogenic evolutionary design experiments in which 7 types of wind bracing elements were used (see Figure 19). The generative representations used in these experiments were based on one-dimensional CAs with 7 possible cell values. They are generalized versions of the generative representations studied in the previous section. A detailed description of this type of generative representations was presented in section 4.4.1.

As before, the experiments were divided into groups: the short-term processes and the long-term processes. In the short-term experiments the optimal values of the generative representation parameters, i.e. the type of CA rules and the radius of the local neighborhood, were sought. On the other hand, the values of the evolutionary computation parameters were assumed based on the results of the short-term morphogenic evolutionary design experiments with elementary CAs (see Table 92) reported in the previous section.

Short-term Morphogenic Evolutionary Design

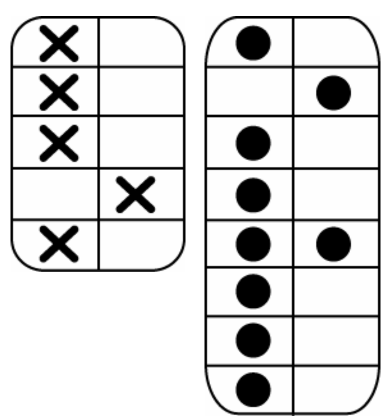


Table 95 shows the generative representation parameters and their values which were used in the short-term morphogenic evolutionary design experiments. As in the previous section, two types of CA rules were investigated: standard and totalistic. In both cases, two lengths of the radius of the local neighborhood were studied experimentally to determine the optimal combination of the generative representation parameters for this design problem.

Table 95. Generative representation parameters and their values used in the short-term morphogenic evolutionary design experiments with 1D cellular automata

Representation Parameter	*Value(s)*
Representation type	Cellular automata
CA dimension	1D
Number of cell states	7
CA rule type	Standard CA rule, or totalistic CA rule
Neighborhood radius	1, or 2
Boundary conditions	Periodic
Design embryo location	Bottom
Design embryo initialization	Random

The lengths of genomes encoding the generative representations of wind bracing systems with 7 types of wind bracing elements were quite different for standard CA rules and totalistic CA rules. In the former case, they consisted of 348 and 16,812 genes for the radius of the local neighborhood equal to 1 and 2, respectively. In the latter case, the corresponding lengths were equal to 24 and 36 genes. In all cases, the genomes were homogenous, i.e. all genes had the same number of possible values (i.e. 7, encoded as integers from 0 to 6) which represented various types of wind bracing elements (see Figure 19).

Optimal Generative Representation Parameters

Figure 119 compares the average best-so-far fitness curves obtained in the short-term morphogenic evolutionary design experiments with 7 types of wind bracing elements. It clearly shows that totalistic CA rules significantly outperformed standard CA rules in this problem domain. The overall best results were produced by totalistic CA rules with the radius of the local neighborhood equal to 2. The average end-of-run fitness obtained in this experiment was equal to 449,194 lbs. and was more than 45,500 lbs., or 9.2 percent, better than the average fitness produced by standard CA rules (with the radius = 1). The difference between the average end-of-run fitness produced by totalistic CA rules with two different radii of the local neighborhood was small and equaled about 2,800 lbs., or 0.6 percent. Thus, both design experiments with totalistic CA rules produced comparable results.

Performance Improvement

Figure 120 compares the results of the short-term morphogenic evolutionary design experiments (standard CA rules and totalistic CA rules) to the ones produced in the evolutionary optimization experiments (see section 7.2.2). It shows that the average best-so-far fitness of the design concepts generated by standard CA rules and totalistic CA rules was better than the average best-so-far fitness obtained in the evolutionary optimization processes. However, the results produced by standard CA rules exhibited significantly higher variance (by an order of magnitude) than the results of the other two design experiments. The overall best results were produced by totalistic CA rules. The average performance improvement achieved in this case exceeded 56,000 lbs., or 11 percent.

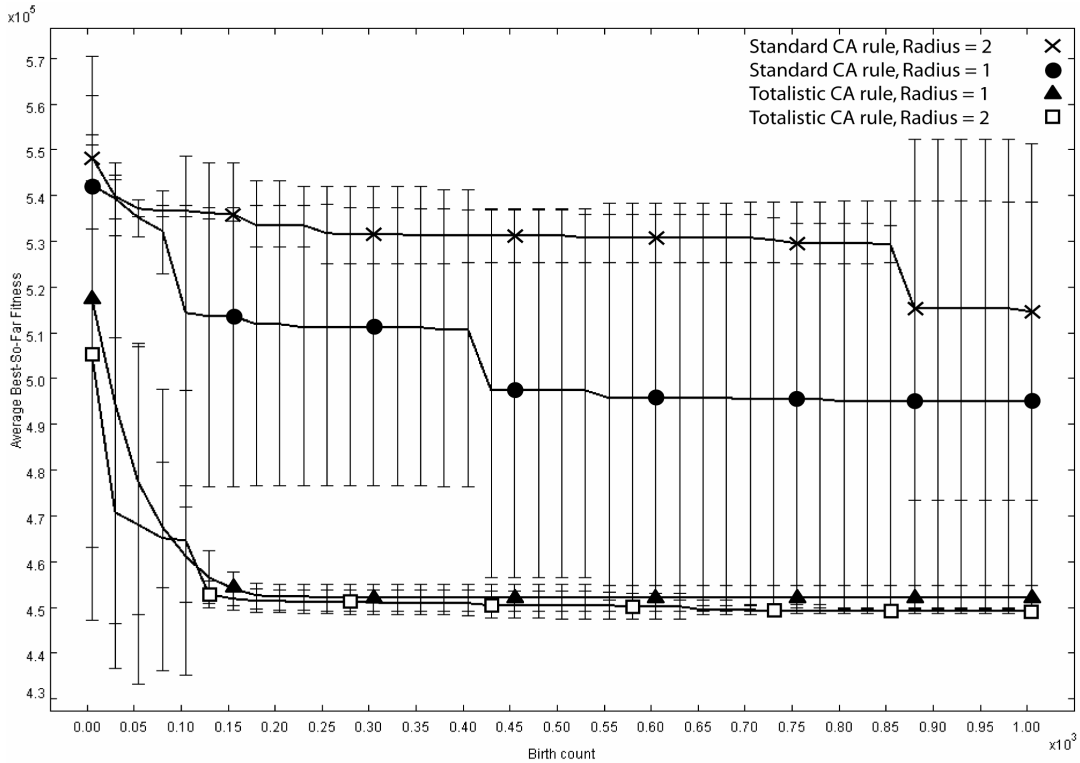


Figure 119. Comparison of the average best-so-far fitness produced in the short-term morphogenic evolutionary design experiments with 7 types of wind bracing elements

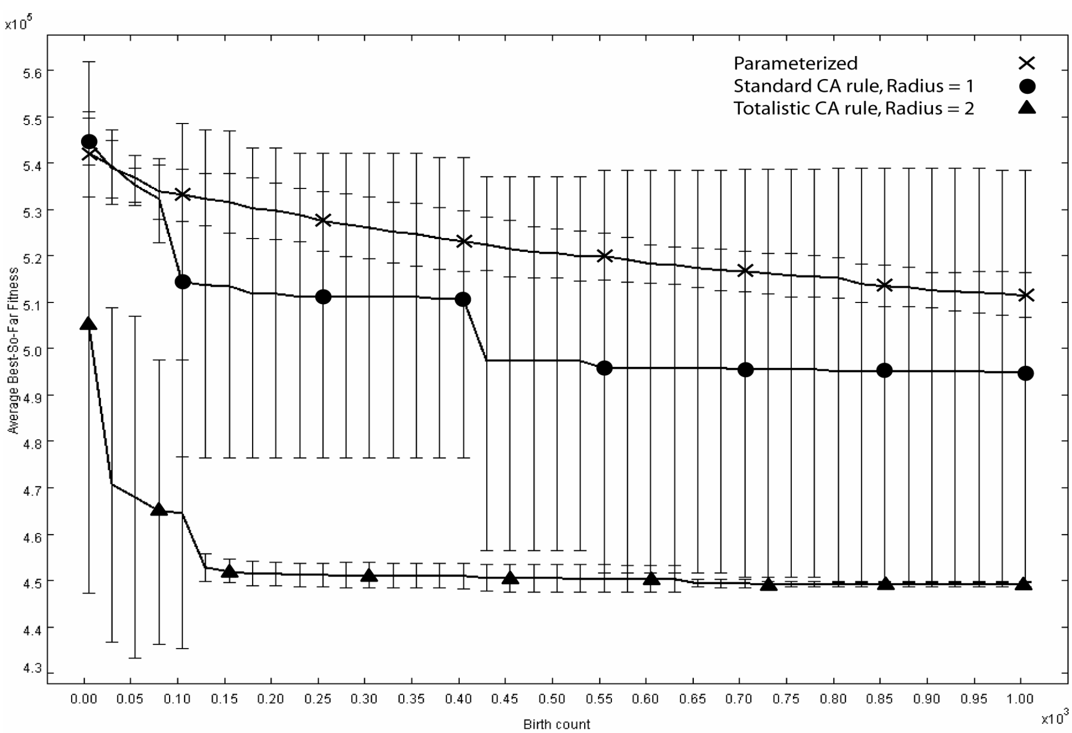
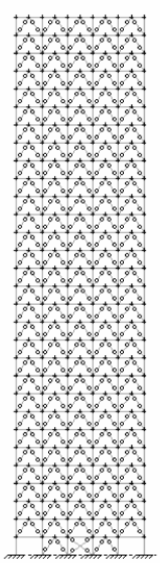
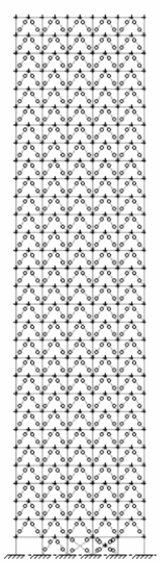
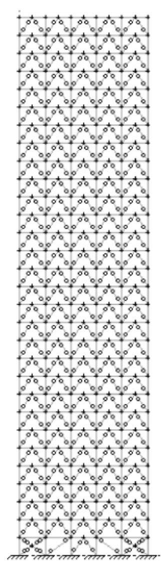
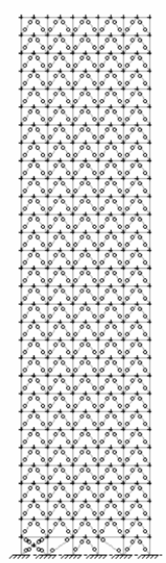


Figure 120. Comparison of the average best-so-far fitness produced in the short-term evolutionary optimization experiments (parameterized representations) and morphogenic evolutionary design experiments with 7 types of wind bracing elements

Optimal Designs

Table 96 presents the best design concepts generated in the short-term morphogenic evolutionary design experiments. The fitness of the best design was equal to 448,597 lbs. It was the best design of a wind bracing system found so far. In fact, all design concepts shown in Table 96 were better than the best design concept produced in the evolutionary optimization experiments (see Table 70 in chapter 7) and the best design concept produced in the experiments with 1D cellular automata (see Table 43 in chapter 6). All design concepts presented in Table 96 exhibit the fully braced pattern consisting of K bracings. The differences among them occur only in the configurations of the design embryo. The fitness of all design concepts shown in Table 96 was improved by introduction of a single or several simple X bracings in the configuration of the design embryo.

Table 96. Best design concepts of wind bracing systems generated in the short-term morphogenic evolutionary design experiments with 7 types of wind bracing elements

			
448,597 5.0203	448,646 5.0196	448,845 5.0246	448,899 5.0249

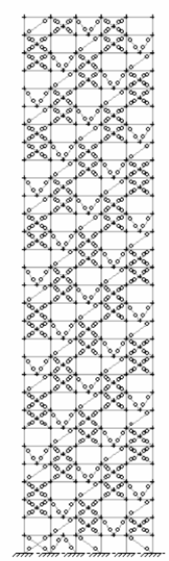
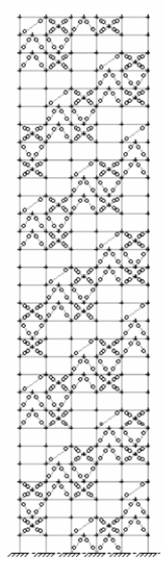
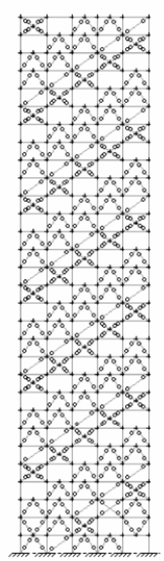
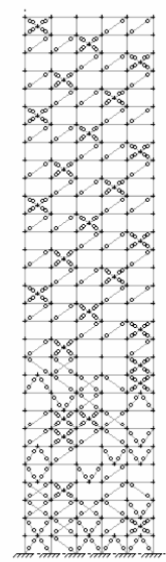
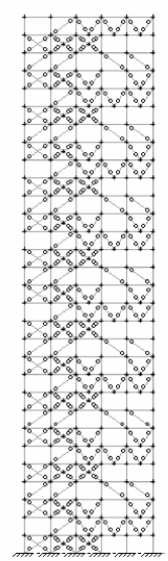
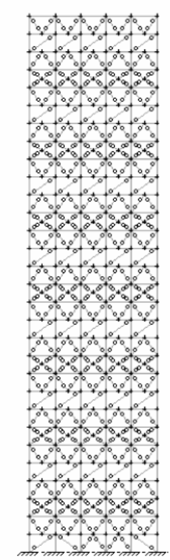
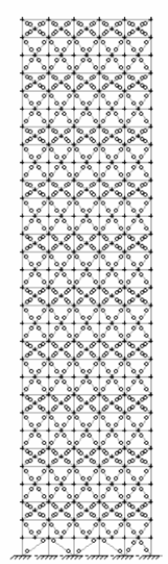
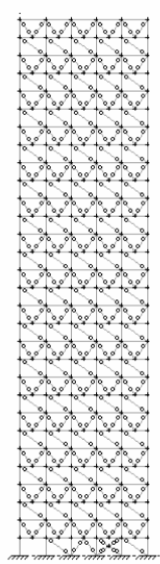
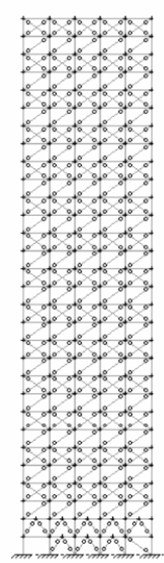
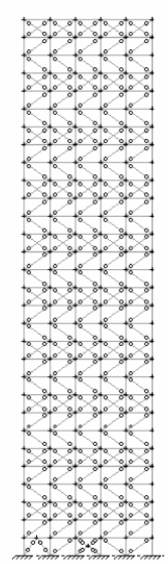
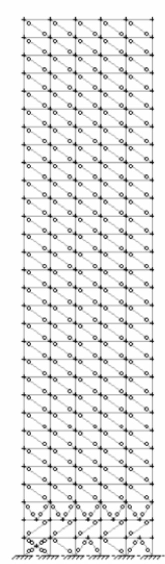
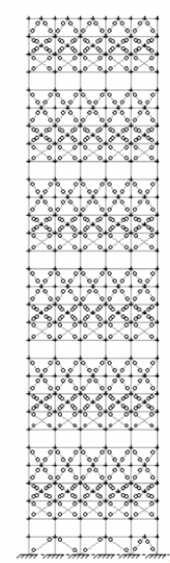
Structural Shaping Patterns

The morphogenic evolutionary design processes with 7 types of wind bracing elements generated many interesting structural shaping patterns. Several examples of such patterns are shown in Table 97. The first 5 patterns were generated by standard CA rules. Majority of them can be classified as elaborate versions of the macro bracing pattern utilizing several types of wind bracing elements. The remaining 7 structural shaping patterns were produced by totalistic CA rules.

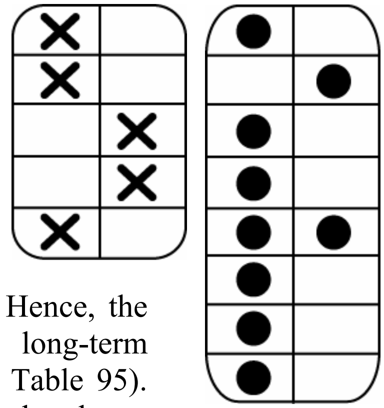
Table 97 clearly shows that there are substantial qualitative differences among the patterns produced by these two types of CA rules. Standard CA rules generate more sophisticated macro bracings patterns or checkerboard patterns which propagate throughout the structural system. Totalistic CA rules, on the other hand, produce fairly uniform or periodic patterns in which the entire configurations of individual stories are occupied by a single type of wind bracings.

In the next subsection, I will investigate the long-term experiments and test if and how much they can improve the performance of morphogenic evolutionary design processes. I will also compare the results of the long-term and the short-term morphogenic evolutionary design experiments as well as results of the long-term evolutionary optimization processes with 7 types of wind bracing elements (see section 7.2.2).

Table 97. Interesting structural shaping patterns generated in the short-term morphogenic evolutionary design experiments with 7 types of wind bracing elements

 520,665 5.6186	 519,650 6.0869	 527,403 5.2536	 532,381 5.9625	 529,715 5.6060	 531,725 5.1872
 521,844 7.4764	 534,237 5.9670	 588,467 4.8697	 585,146 4.3885	 549,902 6.5988	 728,913 5.2248

Long-term Morphogenic Evolutionary Design



The short-term experiments with 1D CAs showed that generative representations with both standard and totalistic CA rules outperformed parameterized representations of wind bracing systems with 7 types of wind bracing elements. There were, however, significant differences between the two types of CA rules in terms of the variance and the average best-so-far fitness. The performance of both types of CA rules was further investigated in the long-term experiments and compared to the results of the long-term evolutionary optimization experiments. Hence, the same generative representation parameters were used in the long-term experiments as the ones used in the short-term experiments (see Table 95). The only exception was that only one radius of the local neighborhood was investigated for each type of CA rules. Standard CA rules were evolved with the radius equal to 1 whereas totalistic CA rules used the radius equal to 2. This choice was motivated by the results of the short-term morphogenic evolutionary design experiments reported in the previous subsection. Evolutionary computation parameters used in the long-term experiments were exactly the same as the ones used in the short-term processes (see Table 92).

Figure 121 compares the average best-so-far fitness curves obtained in the two long-term morphogenic evolutionary design experiments with 7 types of wind bracing elements. It clearly shows that totalistic CA rules outperformed standard CA rules by a wide margin. They also exhibited several orders of magnitude smaller variance than standard CA rules. The average fitness of design concepts after 10,000 evaluations produced using totalistic CA rules was equal to 448,785 lbs. compared to 482,821 lbs. obtained by standard CA rules. Thus, totalistic CA rules outperformed, on average, standard CA rules by more than 34,000 lbs., or 7 percent.

Performance Improvement

Figure 122 compares the results of the long-term morphogenic evolutionary design experiments (totalistic CA rules with the radius equal to 2) to the ones produced in the evolutionary optimization experiments (see section 7.2.2). It shows that the average best-so-far fitness of the design concepts generated by totalistic CA rules was far better than the average best-so-far fitness obtained in the evolutionary optimization processes. Also, the results produced by totalistic CA rules exhibited significantly smaller variance than the results of the evolutionary optimization experiments. The average performance improvement achieved in the long-term morphogenic evolutionary design processes exceeded 56,500 lbs., or 11.2 percent. It was only slightly better (by about 500 lbs. or 0.1 percent) than the average performance improvement obtained in the short-term morphogenic evolutionary design experiments with totalistic CA rules.

At the same time, the long-term morphogenic evolutionary design processes outperformed the long-term evolutionary optimization processes by more than 42,500 lbs., or 8.6 percent. Thus, the performance improvements achieved in the long-term morphogenic evolutionary design experiments with 7 types of wind bracing elements were similar to the ones obtained in the long-term morphogenic evolutionary design processes with 2 types of wind bracing elements and the group No.2 (K bracings).

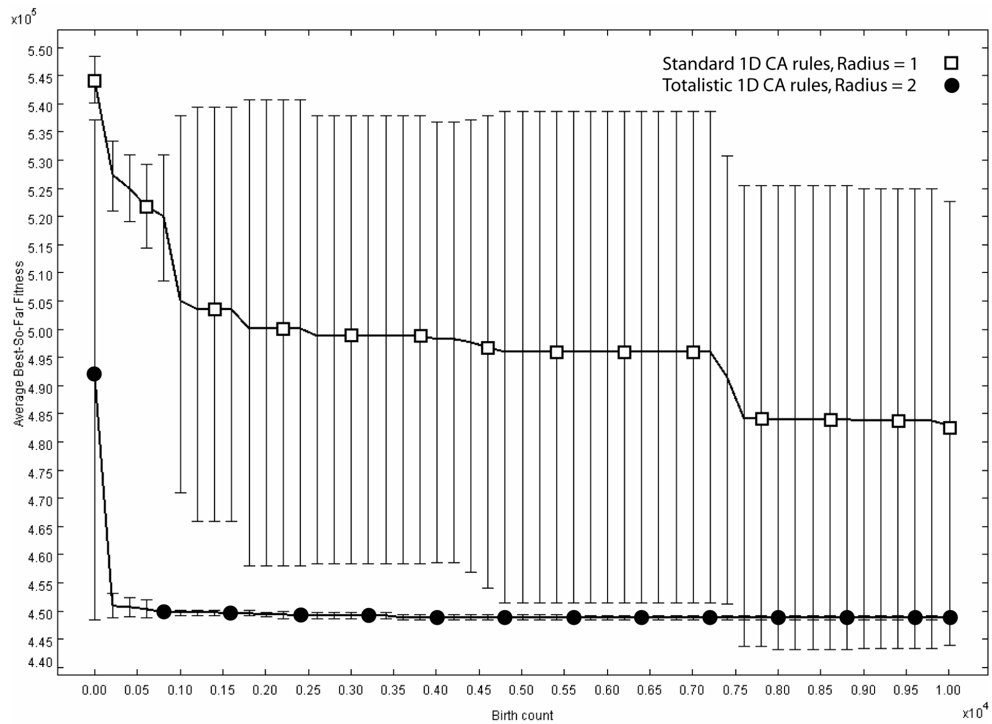


Figure 121. Comparison of the average best-so-far fitness produced in the long-term morphogenic evolutionary design experiments with 7 types of wind bracing elements

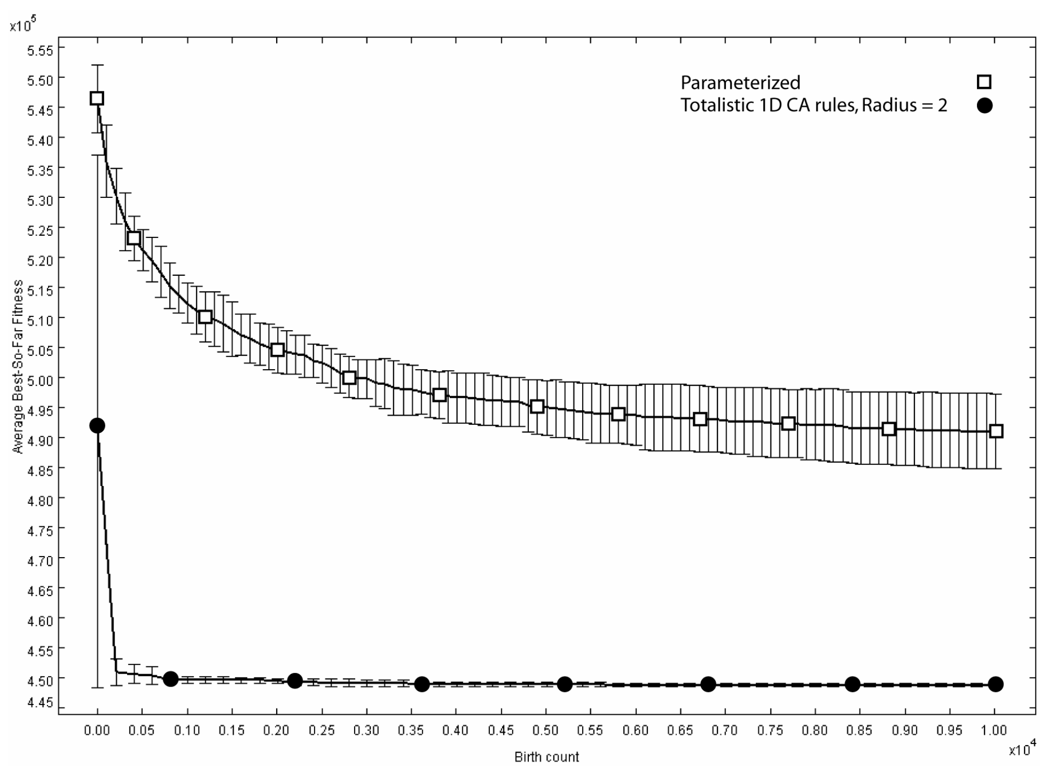


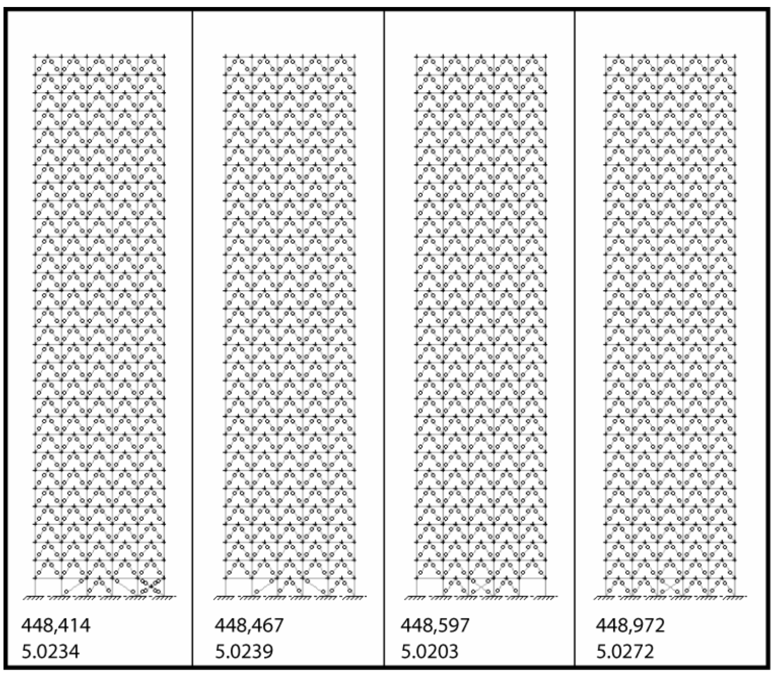
Figure 122. Comparison of the average best-so-far fitness produced in the long-term evolutionary optimization experiments (parameterized representations) and morphogenic evolutionary design experiments with totalistic CA rules and 7 types of wind bracing elements

Optimal Designs

The best designs produced in the long-term morphogenic evolutionary design experiments with 7 types of wind bracing elements are presented in Table 98. The fitness of the best design was equal to 448,414 lbs. It was the best design of a wind bracing system (as far as the total weight of the structural system is concerned) found in the design experiments reported in this dissertation. It was slightly better (by about 180 lbs.) than the best design concept found in the short-term morphogenic evolutionary design processes (see Table 96). It also outperformed the best design concept produced in the long-term evolutionary optimization experiments (see Table 70 in chapter 7) by about 36,600 lbs., or 7.5 percent, and the best design concept produced in the experiments with 1D CAs (see Table 43 in chapter 6) by 670 lbs., or 0.15 percent.

Similarly as in the previous subsection, all design concepts presented in Table 98 exhibit the fully-braced pattern consisting of K bracings. The differences among them occur only in the configurations of the design embryo. Here, the performance improvements were achieved by an introduction of a single simple X bracing, or a combination of two diagonal bracings, into the configurations of the design embryos.

Table 98. Best design concepts of wind bracing systems produced in the long-term morphogenic evolutionary design experiments with 7 types of wind bracing elements



Thus, these results confirm my previous assumption of evolving/optimizing both parts of the generative representation, i.e. the design embryo and the design rule. When the well performing design rules have been found, the morphogenic evolutionary design processes finely tuned the configurations of the design embryos. This resulted in an improved performance of the generated design concepts of wind bracing systems.

Summary

In this section, I described the results of both the short-term and the long-term morphogenic evolutionary design experiments with the generative representations based on one-dimensional

CAs. The number of types of wind bracing elements considered in the experiments was increased to 7 and it included all types of wind bracings shown in Figure 19.

The sensitivity analysis conducted in the short-term experiments focused on the generative representation parameters only. It included the type of the CA rules and the radius of the local neighborhood, i.e. the parameters that had been previously identified as having the biggest impact on the quality of generated design concepts (see chapter 6). On the other hand, the optimal values of the evolutionary computation parameters were assumed based on the results of the short-term morphogenic evolutionary design experiments with elementary CAs (see section 8.2.1)

Both the short-term and the long-term morphogenic evolutionary design experiments have shown that the totalistic CA rules produced significantly better results than the standard CA rules. As in the previous section, they also found the optimal solutions much faster (within 700-800 fitness evaluations). On the other hand, standard CA rules produced more interesting structural shaping patterns.

Figure 123 shows the average performance improvements obtained in the short- and long-term morphogenic evolutionary design experiments with 7 types of wind bracing elements and compares them to the improvements achieved in the experiments with 2 types of wind bracing elements reported in the previous section. It clearly illustrates that the morphogenic evolutionary design processes achieved high levels of performance improvement (more than 11 percent). These results are similar to the ones obtained in the experiments with K bracings (see section 8.2.1). Also, the performance improvements achieved in the short-term experiments and the long-term experiments are almost identical which means that the optimial solutions were produced in the early stages of the morphogenic evolutionary design processes, i.e. the optimal solutions were found quickly.

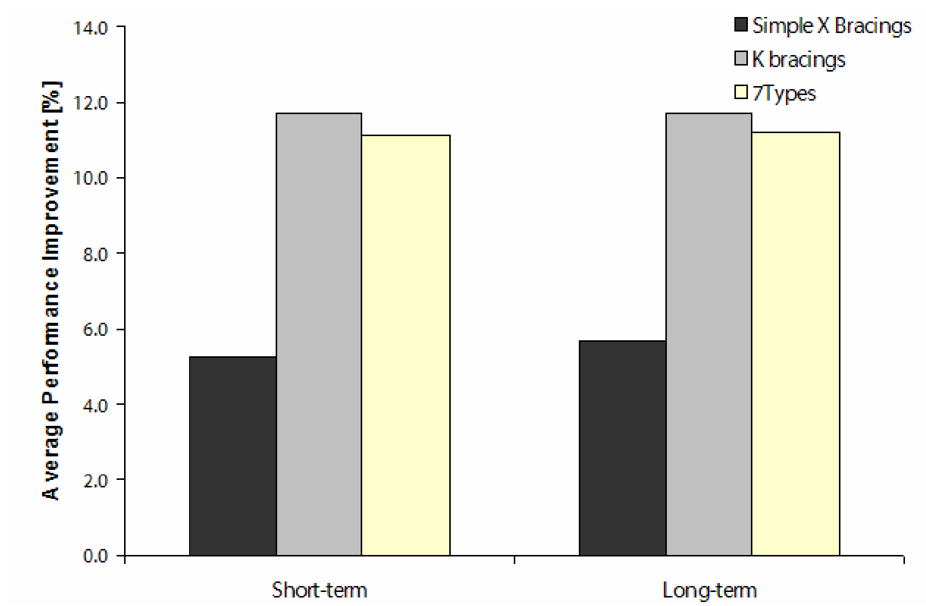


Figure 123. Comparison of the average performance improvements produced in the morphogenic evolutionary design of wind bracing systems with 2 and 7 types of bracing elements in the short-term and long-term experiments

As in the previous section, I compared the results of the short-term and the long-term morphogenic evolutionary design experiments to the results obtained in the evolutionary optimization experiments (see chapter 7). Figure 124 shows the average performance improvements between the morphogenic evolutionary design and the evolutionary optimization achieved in the conducted experiments. It also compares them to the corresponding values obtained in the experiments with 2 types of wind bracing elements reported in the previous section. Figure 124 clearly shows that morphogenic evolutionary design processes significantly outperformed the evolutionary optimization processes. The obtained performance improvement levels were even higher than in the case of morphogenic evolutionary design experiments with K bracings. They exceeded 12 percent and 8 percent in the short-term and the long-term processes, respectively.

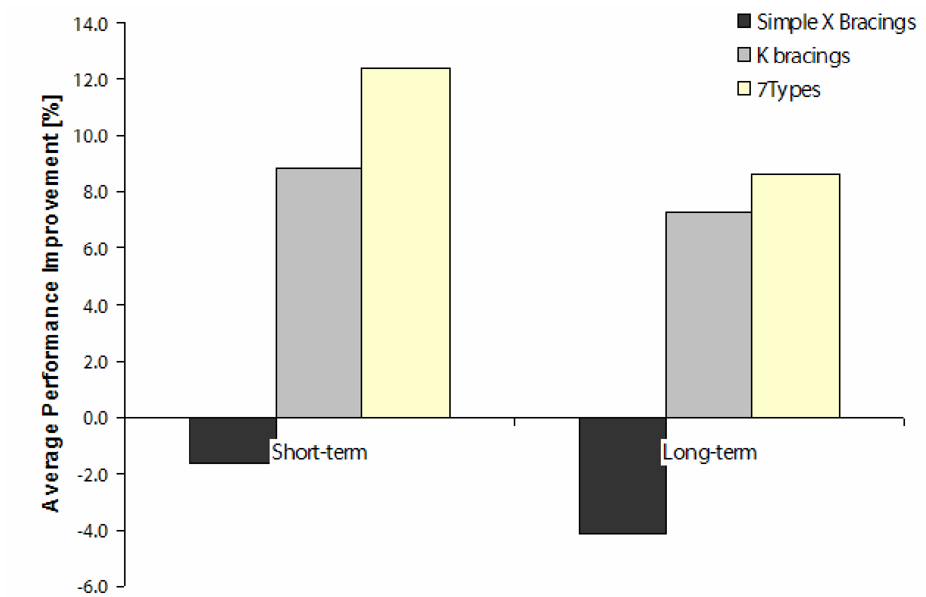


Figure 124. Comparison of the average performance improvements produced in the morphogenic evolutionary design and evolutionary optimization of wind bracing systems with 2 and 7 types of bracing elements in the short-term and long-term experiments

Finally, Figure 125 shows the performance improvements between the best design concepts of wind bracing systems produced in morphogenic evolutionary design experiments with 2 and 7 types of wind bracing elements and the best designs generated by elementary and one-dimensional CAs (see sections 6.2 and 6.3). It shows that the morphogenic evolutionary design experiments with 7 types of wind bracing elements produced only slightly worse designs (by about 0.02 percent) in the short-term experiments. However, in the long-term experiments they generated better design concepts (by about 0.1 percent) than one-dimensional CAs.

In the next section, I will further investigate morphogenic evolutionary design of wind bracing systems in tall buildings. This time, however, 2D cellular automata and 2D design embryos will be studied empirically. Particular emphasis will be put on explicit modeling the local interactions among the structural members using various shapes and radii of the local 2D neighborhood.

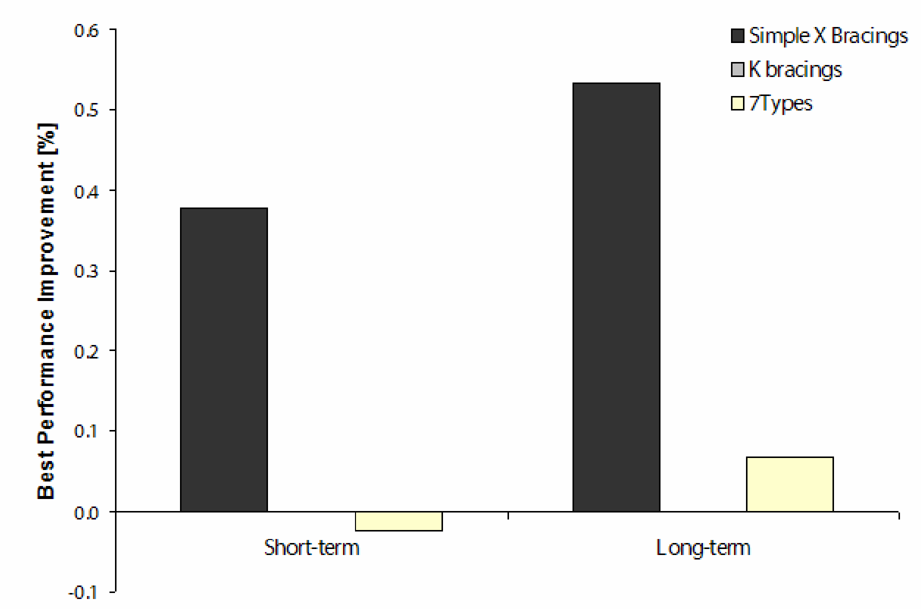
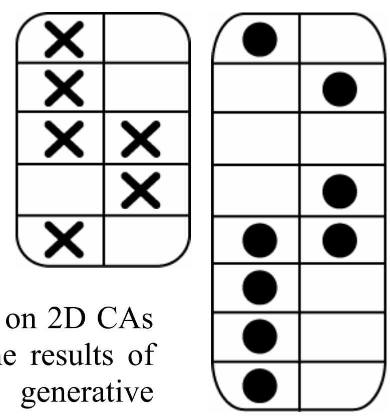


Figure 125. Comparison of the performance improvements between the best designs produced in the morphogenic evolutionary design experiments with 2 and 7 types of wind bracing elements and the best designs generated by elementary and one-dimensional CAs

8.2.3. Evolution of 2D Cellular Automata



So far, morphogenic evolutionary design processes involved only generative representations based on one-dimensional CAs. But as I discussed it in section 6.4, the structural systems considered in this dissertation are inherently two-dimensional. The planar interactions among structural elements cannot be explicitly modeled using 1D CAs. Hence, generative representations based on two-dimensional cellular automata (2D CAs) were proposed in section 4.4.3 to account for these interactions. Initial studies with design concept generators based on 2D CAs were reported in section 6.4. In this section, I will describe the results of morphogenic evolutionary design experiments in which the generative representations based on 2D CAs were evolved by evolutionary algorithms.

Similarly as it was the case with 1D CAs, two types of 2D CAs rules were investigated: standard and totalistic. Besides, in the case of generative representations based on 2D CAs, one has to specify not only the radius of the local neighborhood (2D neighborhood in this case) but also its shape. As in section 6.4, five shapes of the local neighborhood were studied, including Moore, von Neumann, diagonal, north-south, and east-west neighborhoods (see Figure 59). One length of the radius of the local neighborhood (equal to 1) was used in the experiments with standard 2D CA rules while the experiments with totalistic 2D CA rules considered two lengths of the radius (equal to 1 and 2).

The generative representation parameters and their values used in the experiments reported in this section are presented in Table 99. It shows that 2D CAs with 3 possible cell values were used similarly to the design experiments with design concept generators based on 2D CAs (see section 6.4). The values corresponded to 3 types of wind bracing elements: no bracings (empty cells), X bracings, and K bracings. The design embryos had a form of 2D configurations of wind bracing elements and they were initialized randomly in all design experiments reported in this section. The design rules based on 2D CA rules were applied to these 2D design embryos and iterated 50 times. Hence, the value of the *iteration_max* parameter was equal to 50. A detailed description of this type of generative representation and the developmental process during which the design concepts of wind bracing systems were produced from the design embryos is presented in section 4.4.3.

Table 99. Generative representation parameters and their values used in the morphogenic evolutionary design experiments with 2D CAs

Representation Parameter	*Value(s)*
Representation type	Cellular automata
CA dimension	2D
Number of cell states	3
CA rule type	Standard CA rules, or totalistic CA rules
Neighborhood radius	1, or 2
Boundary conditions	Periodic
Shape of the local neighborhood	Moore, von Neumann, diagonal, north-south, or east-west
Design embryo initialization	Random

On the other hand, Table 100 presents the evolutionary computation parameters and their values used in the design experiments. It shows that the values of the evolutionary computation parameters were selected based on the results of morphogenic evolutionary design experiments reported in the previous sections.

Thus, no sensitivity analyses were conducted in the case of evolutionary computation parameters but the most successful values of these parameters determined in the previous morphogenic evolutionary design experiments were employed. Hence, only the long-term experiments were conducted with the generative representations based on 2D CAs and they involved 10,000 fitness evaluations.

The obtained results are reported in the following subsections.

Optimal Type of 2D CA Rules

In this group of experiments, I investigated the impact of the type of the 2D CA rules on the fitness of produced design concepts. As discussed above, two types of 2D CA rules were employed in these experiments: standard 2D CA rules and totalistic 2D CA rules. Figure 126 shows typical results obtained in these experiments. Specifically, the results of two design experiments are presented, each involving the diagonal neighborhood and the radius of the local neighborhood equal to 1.

Table 100. Evolutionary computation parameters and their values used in the morphogenic evolutionary design experiments with 2D CAs

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies (ES)
Generational model	Overlapping for ES(μ+λ)
Population sizes (parent, offspring)	(5,25)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	0.1
Crossover (type, rate)	(uniform, 0.2)
Fitness	Total weight of the steel structure (determined by the first-order structural analysis)
Initialization method	Random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	10,000 fitness evaluations
Number of runs	5 in each experiment

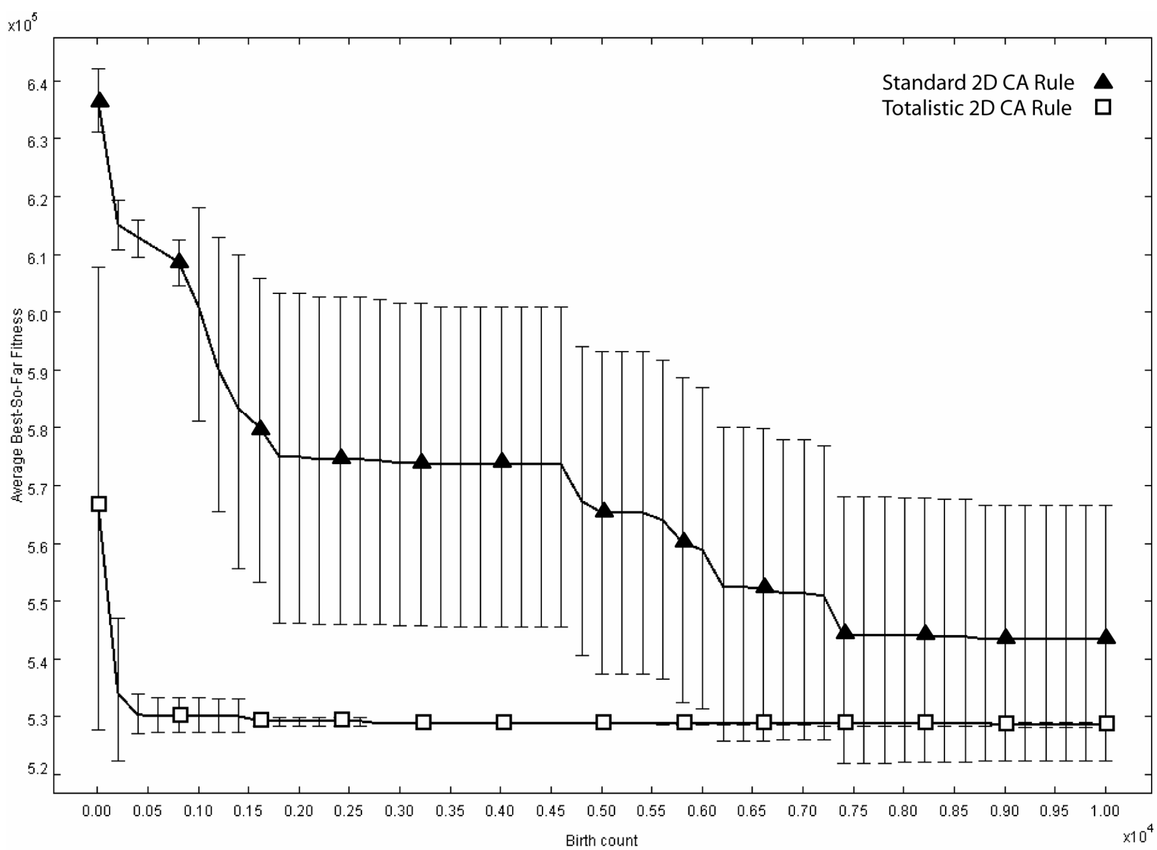


Figure 126. Impact of the type of a 2D CA rule on the average best-so-far fitness of the produced design concepts of wind bracing systems

The results shown in Figure 126 are similar to the ones observed in the morphogenic design experiments with 1D CAs (see for example Figure 119). Specifically, morphogenic evolutionary design processes with totalistic 2D CA rules produced design concepts of better fitness than the ones with standard 2D CA rules. At the same time, totalistic 2D CA rules found the optimal design concepts much faster than standard 2D CA rules. Figure 126 also shows that there were significant differences between the variances generated in these two morphogenic evolutionary design processes. Standard 2D CA rules produced results with much higher variance than totalistic 2D CA rules. Thus, the results obtained in the morphogenic evolutionary design experiments with standard and totalistic 2D CA rules were consistent with my previous findings described in sections 8.2.1 and 8.2.2.

Optimal Shape of Local 2D Neighborhood

In this group of experiments, the impact of the shape of the local neighborhood on the average best-so-far fitness of morphogenic evolutionary design processes was investigated. As discussed above, 5 shapes of the local neighborhood were studied experimentally, including Moore, von Neumann, diagonal, north-south, and east-west neighborhoods.

Figure 127 illustrates the average best-so-far fitness curves obtained in the morphogenic evolutionary design experiments with 4 different shapes of the local neighborhood (the east-west neighborhood produced virtually identical results as the north-south neighborhood and hence was omitted in this figure). It clearly shows that the differences among the curves occurred only in the initial stages of evolution up to 1,500 fitness evaluations (see the zoom in window on the left hand side in Figure 127). At the end of the long-term design processes all shapes of the local neighborhood produced comparable results.

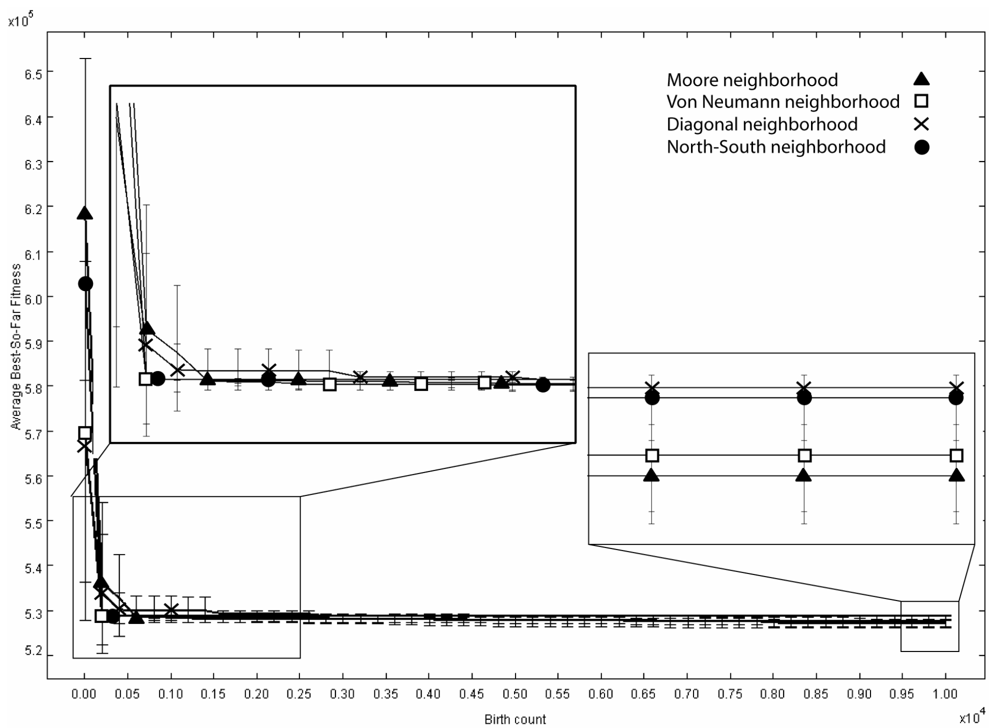


Figure 127. Impact of the shape of the local neighborhood on the average best-so-far fitness of design concepts of wind bracing systems produced by morphogenic evolutionary design experiments with 2D CAs

The fastest progress in the initial stages of evolution was achieved when von Neumann neighborhood and the north-south neighborhood were used. On the other hand, 2D CA rules with Moore neighborhood achieved the smallest progress in the initial part of the morphogenic evolutionary design process but at the end of the run generated the overall best results (see the zoom in window on the right hand side in Figure 127). However, the differences among the end-of-run fitness produced by various shapes of the local neighborhood were small and not statistically significant.

In the next subsection, I will investigate another important parameter defining the extent of the interactions in the local neighborhoods, namely the radius of the local neighborhood.

Optimal Radius of Local 2D Neighborhood

In this group of experiments, the impact of the radius of the 2D local neighborhood on the average best-so-far fitness of generated design concepts was investigated. As described earlier, two sizes of the radius of the local neighborhood were studied for totalistic 2D CA rules.

Figure 128 shows typical results regarding the impact of the length of the radius on the progress of morphogenic evolutionary design processes. In this case, 2D CA rules with the diagonal neighborhood were employed and two lengths of the radius investigated, namely 1 and 2. Figure 128 clearly shows that the differences in the performance were limited only to the initial stages of evolution, similarly as it was the case with the shape of the local neighborhood discussed in the previous subsection. After about 1,500 fitness evaluations the two morphogenic evolutionary design processes produced almost identical results in terms of the average best-so-far fitness of generated design concepts as well as its variance.

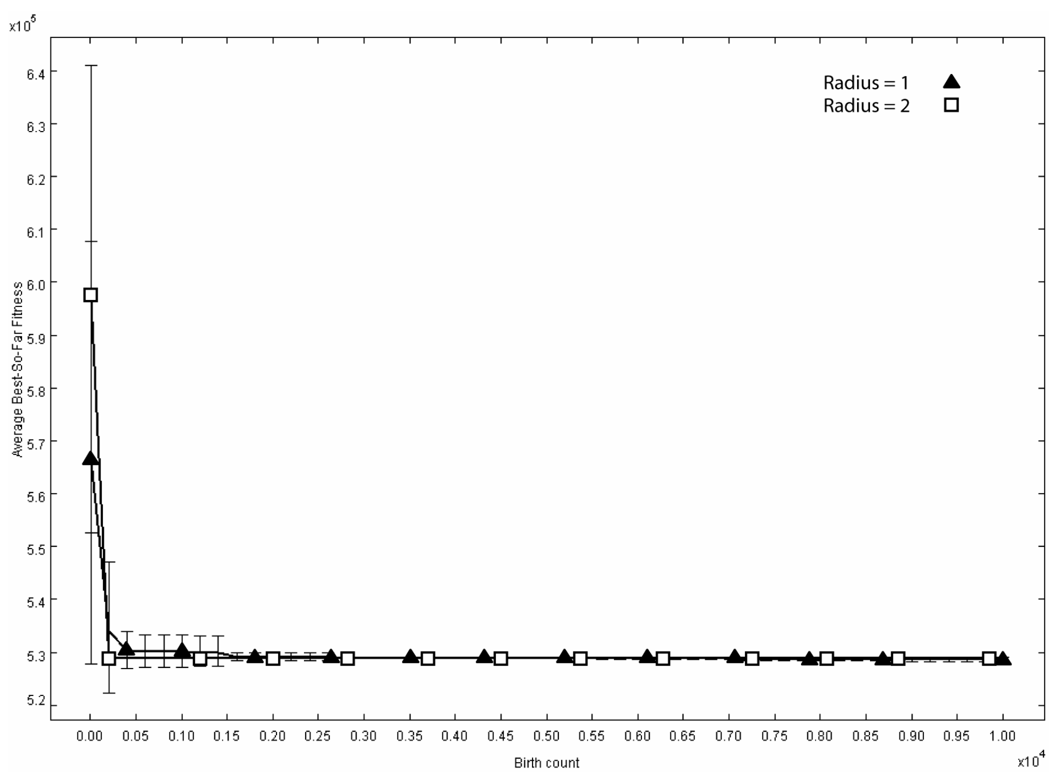


Figure 128. Impact of the radius of the local 2D neighborhood (here the diagonal neighborhood) on the average best-so-far fitness of design concepts of wind bracing systems produced by morphogenic evolutionary design experiments with 2D CAs

Summary

In this section, I described the results of the long-term morphogenic evolutionary design experiments with 2D CAs. The experiments reported in this section focused on modeling the local interactions among structural members using various shapes and radii of the local 2D neighborhoods and on testing their impact on the progress of morphogenic evolutionary design processes. Hence, I investigated two important generative representation parameters which define the local neighborhoods in 2D CAs: the shape of the local neighborhood and the radius of the local neighborhood. Besides, the impact of the type of CA rules (standard 2D CAs vs. totalistic 2D CAs) on the performance of morphogenic evolutionary design processes was also studied.

The experimental results presented in the first subsection confirmed my previous findings (see sections 8.2.1 and 8.2.2) regarding the impact of the type of CA rules on the performance of morphogenic evolutionary design processes. Totalistic 2D CA rules outperformed standard 2D CA rules both in producing design concepts of better fitness and in finding these optimal solutions faster. They also showed significantly smaller variance.

The remaining two subsections investigated the parameters which define the shape and size of the local neighborhoods. The conducted experiments showed that the impact of these two parameters on the progress of morphogenic evolutionary design processes was limited to the initial stages of evolution. Specifically, 2D CA rules with von Neumann and north-south neighborhoods found better solutions faster than the 2D CA rules with other shapes of the local neighborhood. Similarly, 2D CA rules with a longer radius of the local neighborhood found optimal solutions faster than the 2D CA rules with the smaller radius. The average end-of-run results were, however, comparable for all shapes and radii of the local neighborhood and no significant differences could be identified.

8.2.4. Summary

In this section, I described the results of morphogenic evolutionary design experiments involving various types of generative representations of wind bracing systems in tall buildings. These generative representations were evolved by evolutionary algorithms in order to find optimal design rules and optimal design embryos which subsequently developed optimal design concepts. All types of representations considered in this section were introduced earlier in chapter 4.

In the first subsection, I described the results of the morphogenic evolutionary design experiments in which the simplest generative representations based on elementary CAs were studied. The number of types of wind bracing elements considered in these experiments was limited to 2 only. Even these simple generative representations produced design concepts of very good performance. They also exhibited interesting structural shaping patterns which were qualitatively different than the patterns produced in the evolutionary optimization experiments (see chapter 7). Initial sensitivity studies conducted during the short-term experiments have shown that optimal mutation rates for morphogenic evolutionary design porcesses are higher than the ones used in the evolutionary optimization procesees. They also showed that totalistic CAs usually produce better design concepts in this problem domain and at the same time find the optimal solutions faster than standard CAs.

The average performance improvements achieved in the short-term and the long-term experiments varied between 5 and 11 percent depending on the group of wind bracing elements used in the design process (simple X bracings or K bracings). Morphogenic evolutionary design processes also found better design concepts of wind bracing systems than the ones found in the experiments with design concept generators (see section 6.2). They also produced significantly better results than evolutionary optimization experiments when the second group of wind bracing elements was employed, i.e. K bracings. They, however, produced inferior results to the evolutionary optimization processes when the first group of wind bracing elements was used (simple X bracings).

In the second subsection, I considered a more general design problem in which all 7 types of wind bracing elements were used in the design processes. In this case, morphogenic evolutionary design processes employed the generative representations based on one-dimensional CAs. The sensitivity analysis conducted in the short-term experiments showed that two generative representation parameters, i.e. the type of CA rules and the radius of the local neighborhood, have the biggest impact on the quality of generated design concepts. Both the short-term and the long-term morphogenic evolutionary design experiments have shown that totalistic CA rules produced significantly better results than standard CA rules. As in the previous section, they also found the optimal solutions much faster (within 700-800 fitness evaluations). On the other hand, standard CA rules produced more interesting structural shaping patterns.

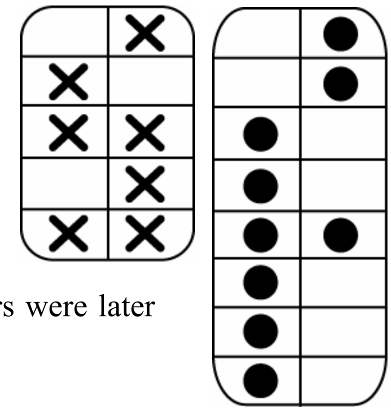
The performance improvements achieved in the morphogenic evolutionary design processes exceeded 11 percent both in the short-term and in the long-term experiments. These processes significantly outperformed evolutionary optimization processes (see section 7.2.2). They also generated better design concepts of wind bracings systems than the design concepts generators studied in section 6.3.

In the final subsection, I reported the results of the long-term morphogenic evolutionary design experiments with 2D CAs. The experiments focused on modeling the local interactions among structural members using various shapes and radii of the local 2D neighborhoods and on testing their impact on the progress of morphogenic evolutionary design processes. The experiments showed that totalistic 2D CA rules produce better design concepts than standard 2D CA rules in this problem domain and they find these optimal solutions faster. Thus, these results were consistent with the findings reported in the previous subsections. The experiments also showed that the impact of the shape and the radius of the local neighborhood is limited to the initial stages of evolution. The end-of-run results were comparable for all shapes and radii of the local neighborhood.

In the experimental part of this dissertation, I also studied the generative representations of wind bracing systems based on multiple 1D CAs (see section 4.4.2). I conducted several morphogenic evolutionary design experiments to test these types of generative representations. The obtained results were, however, significantly inferior to other types of generative representations discussed in this section and hence not included in this chapter. These types of generative representations turned out to be too disruptive to create structural shaping patterns of good performance. Besides, the morphogenic evolutionary design processes involving these representations exhibited very high variance.

In the next section, I will scale up the difficulty of the design problem and consider morphogenic evolutionary design of the entire steel structural systems in tall buildings.

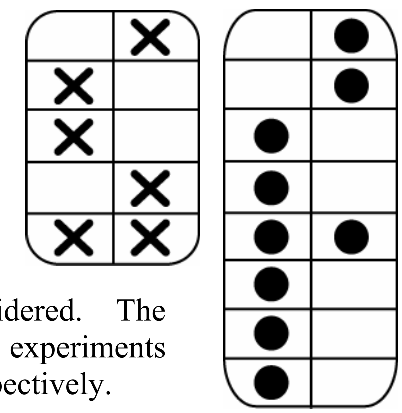
8.3. Morphogenic Evolutionary Design of the Entire Steel Structures



In this section, I will empirically investigate the morphogenic evolutionary design processes in the context of the entire steel structural systems in tall buildings. As before, the experiments reported in this section were divided into two groups: the short-term experiments and the long-term experiments. During the short-term experiments an extensive search of the evolutionary computation parameters and the generative representation parameters was conducted in order to find their optimal values. The optimal values of these parameters were later used in the long-term experiments.

The obtained results are reported in the following subsections.

Short-term Morphogenic Evolutionary Design



In this group of experiments, the short-term morphogenic evolutionary design processes of the entire steel structural systems in tall buildings were conducted. In the experiments, 7 types of wind bracing elements, 2 types of beams, and 2 types of supports were considered. As in the experiments reported in chapter 7, the columns were kept the same during the entire morphogenic evolutionary design processes. Table 101 shows the parameters of the problem investigated in this subsection. Here, 30-story buildings with 6 bays were considered. The heights of stories and bay widths were exactly the same as in the experiments reported in previous sections and equal to 14 feet and 20 feet, respectively.

Table 101. Problem parameters and their values used in the short-term morphogenic evolutionary design of the entire steel structural systems in tall buildings

Problem Parameter	*Value(s)*
Problem type	Design of the entire steel structural system in a tall building
Number of stories	30
Number of bays	6
Bay width	20 feet (6.01 m)
Story height	14 feet (4.27 m)
Distance between transverse systems	20 feet (6.01 m)
Types of bracing elements	No, Diagonal \, Diagonal /, K, V, Simple X, and X
Types of beam elements	Pinned-Pinned, and Fixed-Fixed
Types of column elements	Fixed-Fixed (only)
Types of supports	Pinned and Fixed

The evolutionary computation parameters and their values are presented in Table 102. As in the experiments reported in the previous sections, ES with the overlapping generational model, i.e. ES(μ+λ), were employed. Four combinations of parent and offspring population sizes were studied. Also, nine combinations of mutations and crossover rates were investigated to find their optimal rates. The experiments were repeated 30 times for each combination of mutation and crossover rates, using a different value of the random seed each time. As before, the fitness of the generated design concepts was determined by the total weight of the steel structural system and calculated by the first-order structural analysis.

Table 102. Evolutionary computation parameters and their values used in the short-term morphogenic evolutionary design of the entire steel structural systems

EC Parameter	*Value(s)*
Evolutionary algorithm	Evolution Strategies (ES)
Generational model	Overlapping for ES(μ+λ)
Population sizes (parent, offspring)	(1,25), (1,125), (5,25), or (5,125)
Selection (parent, survival)	(uniform stochastic, truncation)
Mutation rate	1/L, 0.025, 0.1, 0.3, or 0.5
Crossover (type, rate)	(uniform, 0), or (uniform, 0.2)
Fitness	Total weight of the steel structure (determined by the first-order structural analysis)
Initialization method	Random
Constraint handling method	Death penalty (infeasible designs assigned 0 fitness)
Termination criterion	1,000 fitness evaluations
Number of runs	30 in each experiment

The generative representation parameters and their values are presented in Table 103. It shows that both standard CA rules and totalistic CA rules were studied in the experiments reported in this section. Unlike the experiments described in the previous section, only one length of the radius of the local neighborhood was investigated, i.e. the radius equal to 1.

Optimal Mutation and Crossover Rates

The sensitivity analysis conducted during the short-term morphogenic evolutionary design experiments was aimed to determine the optimal rates of mutation and crossover operators as well as parent and offspring population sizes. Typical results regarding the impact of the mutation rate on the progress of evolution are presented in Figure 129. It shows the results of the experiments in which the generative representations based on standard CA rules were used. The rate of the crossover operator was equal to 0.2.

Table 103. Generative representation parameters and their values used in the short-term morphogenic evolutionary design of the entire steel structural systems

Representation Parameter	*Value(s)*
Representation type	Cellular automata
CA dimension	1D
Number of cell states	7 (wind bracings), and 2 (beams)
CA rule type	Standard CA rules, or totalistic CA rules
Neighborhood radius	1
Boundary conditions	Periodic
Design embryo location	Bottom
Design embryo initialization	Random

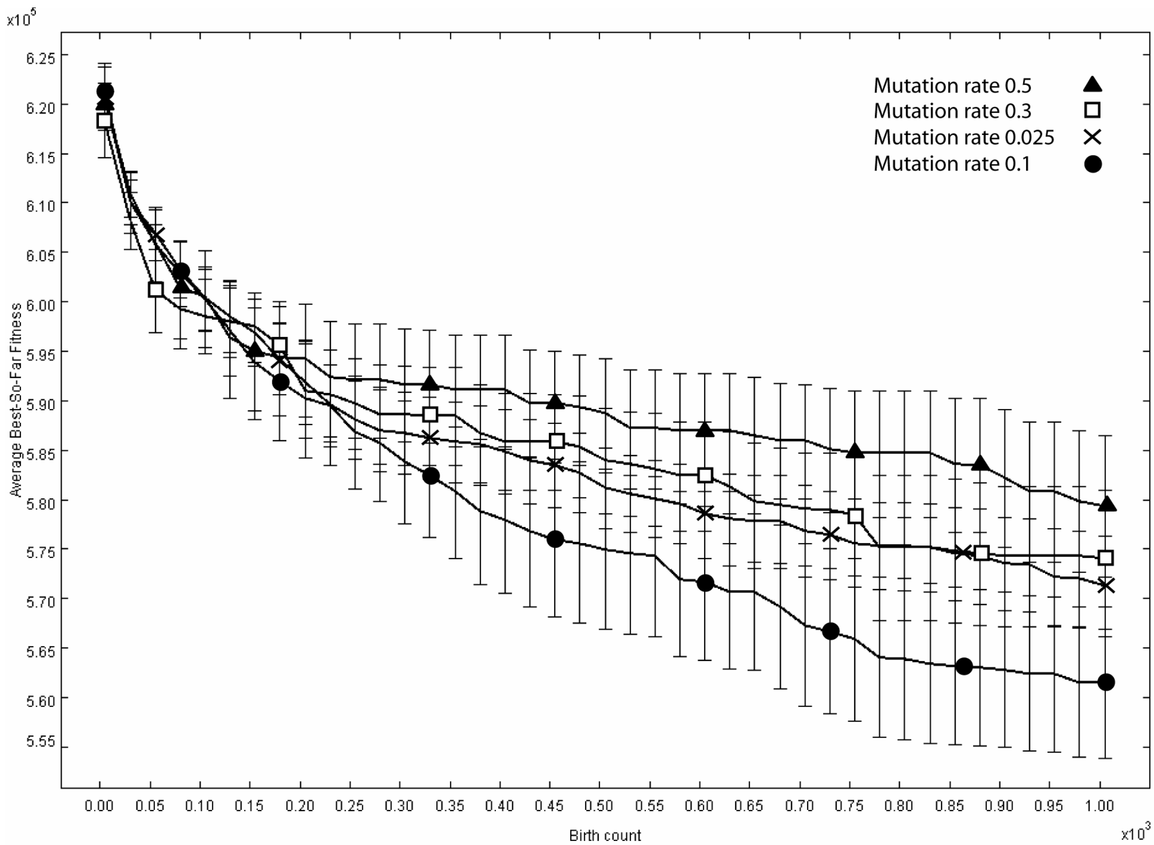


Figure 129. The influence of the mutation rate on the fitness of the design concepts generated in the short-term morphogenic evolutionary design experiments with the entire steel structural systems

Figure 129 shows that the best evolutionary progress was achieved when the mutation rate was equal to 0.1. Similar results were obtained in the design experiments with the generative representations with totalistic CA rules. The impact of the crossover rate was insignificant on the average best-so-far fitness of the produced design concepts. In some cases, better results were produced when the rate was equal to 0.2 and sometimes the best evolutionary progress was achieved when the crossover operator was not applied at all.

Optimal Population Sizes

In another group of experiments, the impact of the sizes of parent and offspring populations on the fitness of produced design concepts was investigated. As it is shown in Table 102, 4 combinations of parent and offspring population sizes were considered: ES(1+5), ES(1+125), ES(5+25), and ES(5+125). Figure 130 shows typical results obtained in these experiments. Here, the generative representations based on totalistic CAs were used with these four combinations of parent and offspring population sizes. All other evolutionary computation parameters, i.e. mutation and crossover rates, were the same in the experiments shown in this figure.

Figure 130 clearly shows that the best results were produced with the parent population size equal to 5 and the offspring population size equal to 25. On the other hand, the worst results were produced by the 'greedy' ES(1+25) in which only the best individual in the population survives to the next generation. Thus, ES(5+25) were subsequently used in the long-term morphogenic evolutionary design of the entire steel structural systems.

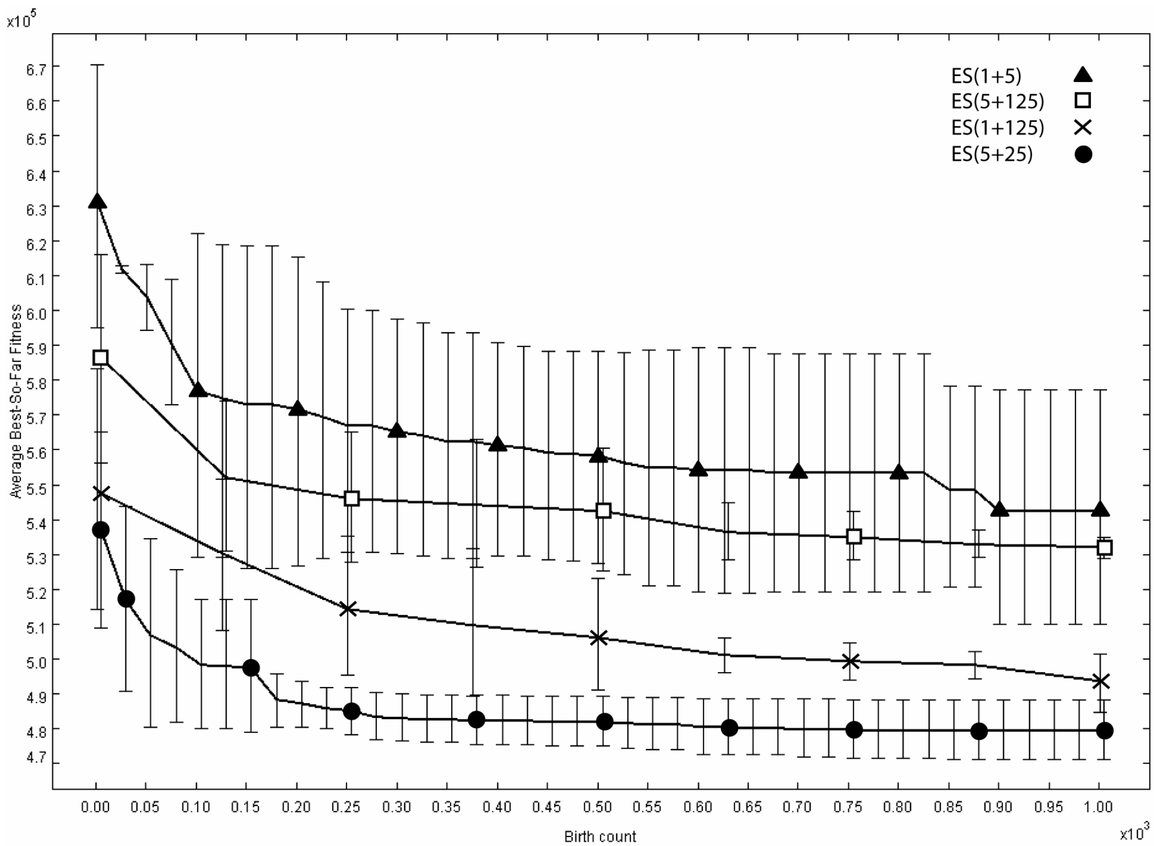


Figure 130. The impact of the parent and offspring population sizes on the fitness of the design concepts generated in the short-term morphogenic evolutionary design of the entire steel structural systems

Optimal Type of CA rules

The sensitivity analysis conducted during the short-term morphogenic evolutionary design experiments also involved the generative representation parameters. Specifically, the impact of the type of the CA rules, i.e. standard vs. totalistic, was investigated. Figure 131 present typical experimental results produced by standard CA rules and totalistic CA rules. It clearly shows that totalistic CA rules outperformed standard CA rules in this problem domain. They also produced the optimal design concepts of the entire structural systems faster than standard CA rules. The results shown in Figure 131 were produced in morphogenic evolutionary design processes in which ES(5+25) was employed and the mutation and crossover rates were equal to 0.1 and 0.2, respectively. The difference between the average best-so-far fitness after 1,000 fitness evaluations between totalistic CA rules and standard CA rules was equal to 35,000 lbs., or 6.2 percent. The results shown in Figure 131 are consistent with the findings reported in the previous section in which the morphogenic evolutionary design of a wind bracing system was considered.

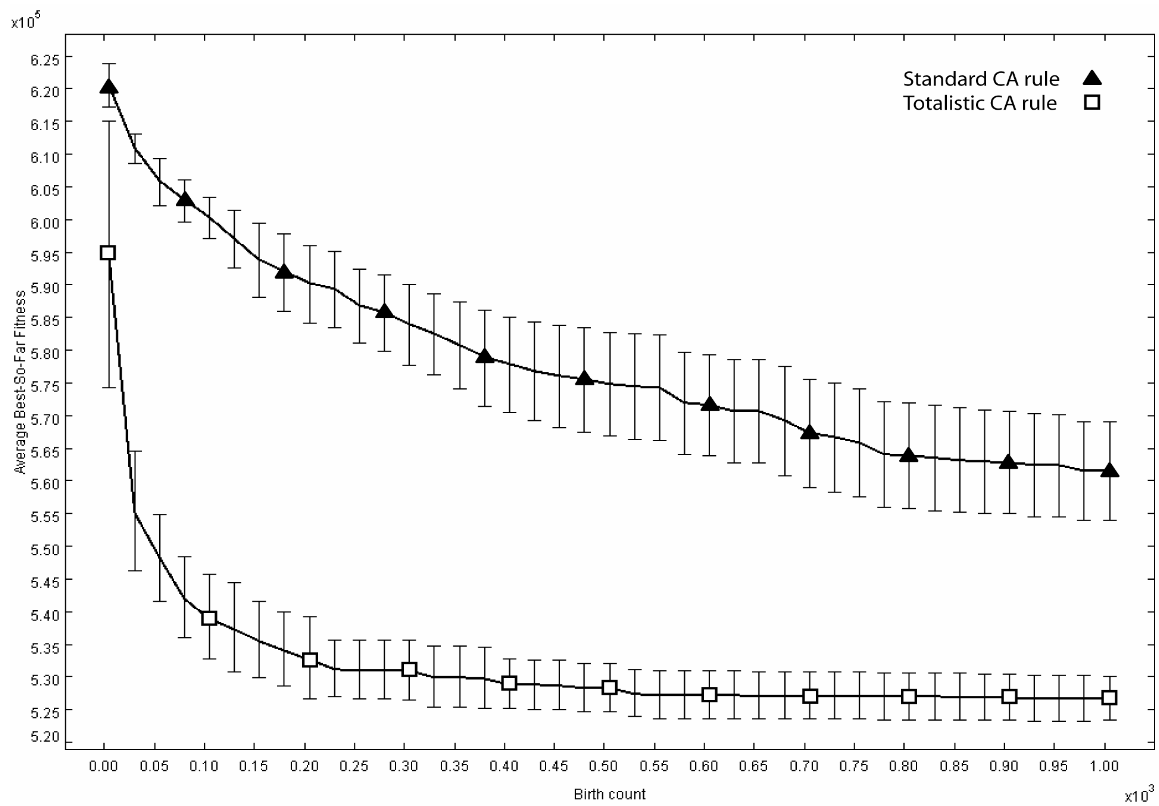


Figure 131. The impact of the type of the CA rule on the fitness of the design concepts generated in the short-term morphogenic evolutionary design of the entire steel structural systems

Performance Improvement

Figure 132 compares the average best-so-far fitness curves produced in the morphogenic evolutionary design processes (standard CA rules and totalistic CA rules) to the results obtained in the evolutionary optimization experiments. It shows that both types of the generative representations significantly outperformed the parameterized representations in this problem domain. The average end-of-run fitness of design concepts produced by totalistic CA rules was equal to 526,592 lbs. It was more than 57,000 lbs., or 9.7 percent, better than the average end-of-run fitness produced by the parameterized representations. Also, standard CA rules generated, on average, better design concepts than the parameterized representations by almost 35,000 lbs., or 6.6 percent. The average performance improvement of the morphogenic evolutionary design processes in the short-term experiments was equal to 68,300 lbs. (11.5 percent) and 59,050 lbs. (9.5 percent) for totalistic CA rules and standard CA rules, respectively. Thus, it was comparable to the average performance improvements achieved in the short-term morphogenic evolutionary design of wind bracing systems (K bracings and 7 types of wind bracings) reported in the previous section.

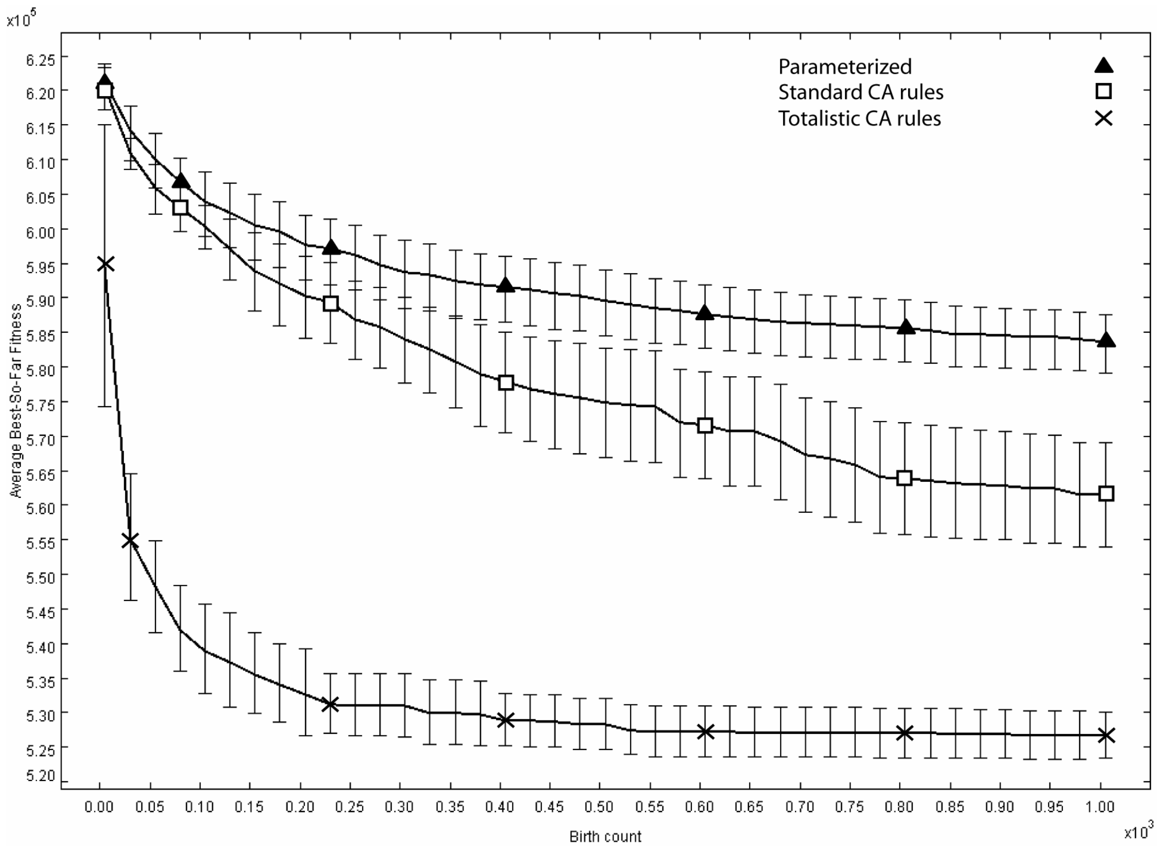
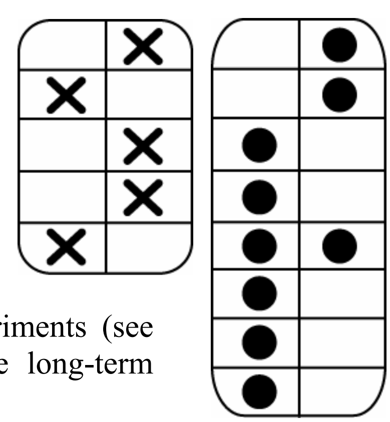


Figure 132. Comparison of the average best-so-far fitness produced in the evolutionary optimization experiments (parameterized representations) and morphogenic evolutionary design experiments with standard CA rules and totalistic CA rules

Long-term Morphogenic Evolutionary Design



The short-term experiments with 1D CAs showed that generative representations with both standard and totalistic CA rules outperformed parameterized representations of the entire steel structural systems in tall buildings. The performance of both types of CA rules was further investigated in the long-term experiments and compared to the results of the long-term experiments with parameterized representations. Hence, the same generative representation parameters were used in the long-term experiments as the ones used in the short-term experiments (see Table 103). Evolutionary computation parameters used in the long-term experiments involved ES(5+25), the mutation rate 0.1, and the crossover rate equal to 0.2.

Figure 133 compares the average best-so-far fitness curves obtained in the two long-term morphogenic evolutionary design experiments with standard and totalistic CAs and compares them to the results produced in the long-term evolutionary optimization experiments. It clearly shows that totalistic CA rules outperformed standard CA also in the long-term processes by about 11,500 lbs., or 2.1 percent. Both long-term morphogenic evolutionary design processes were significantly better than the long-term evolutionary optimization process (parameterized representations). The average fitness of design concepts after 10,000 evaluations produced using totalistic CA rules was equal to 531,830 lbs. compared to 621,425 lbs. obtained in the evolutionary optimization experiment. Also, standard CA rules outperformed the parameterized representations by more than 78,000 lbs., or 12.5 percent. Again, these results are consistent with the previous findings from the morphogenic evolutionary design of wind bracings systems and the short-term morphogenic evolutionary design of the entire steel structural systems.

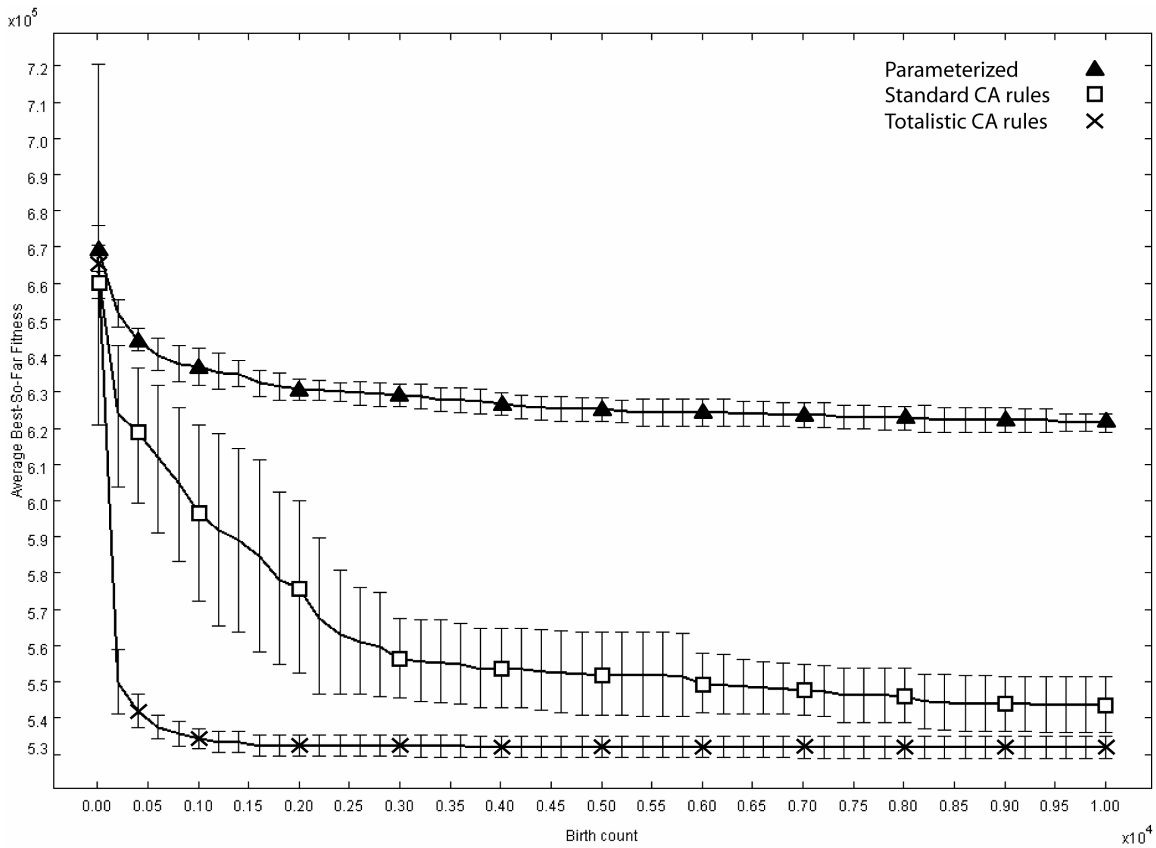


Figure 133. Comparison of the average best-so-far fitness of design concepts of the entire steel structural systems produced in the long-term morphogenic evolutionary design experiments (standard CA rules and totalistic CA rules) and the long-term evolutionary optimization experiments (parameterized representations)

Summary

In this section, I described the results of both the short-term and the long-term morphogenic evolutionary design experiments with the generative representations of the entire steel structural systems in tall buildings. The sensitivity analysis conducted in the short-term experiments focused on the optimal evolutionary computation parameters and one generative representation parameter (the type of the CA rules).

In the short-term experiments, I identified the optimal rates of mutation (equal to 0.1) and the sizes of parent and offspring populations (5 and 25, respectively). The experiments with totalistic and standard CA rules showed that the former significantly outperformed the latter both in the short-term and in the long-term experiments. They also found the optimal solutions much faster.

Figure 134 shows the average performance improvements obtained in the short- and long-term morphogenic evolutionary design experiments with the entire steel structural systems. It clearly illustrates that the morphogenic evolutionary design processes achieved high levels of performance improvement (almost 20 percent in the case of totalistic CA rules). These results were even better than the ones obtained in the experiments with K bracings and 7 types of wind bracings (see sections 8.2.1 and 8.2.2). Also, the performance improvements achieved in the short-term experiments and the long-term experiments are almost identical for the totalistic CA rules which means that the optimial solutions were produced in the early stages of the morphogenic evolutionary design processes. On the other hand, standard CA rules exhibited a steady progress and ultimately produced the performance improvement levels close the the ones achieved by totalistic CA rules.

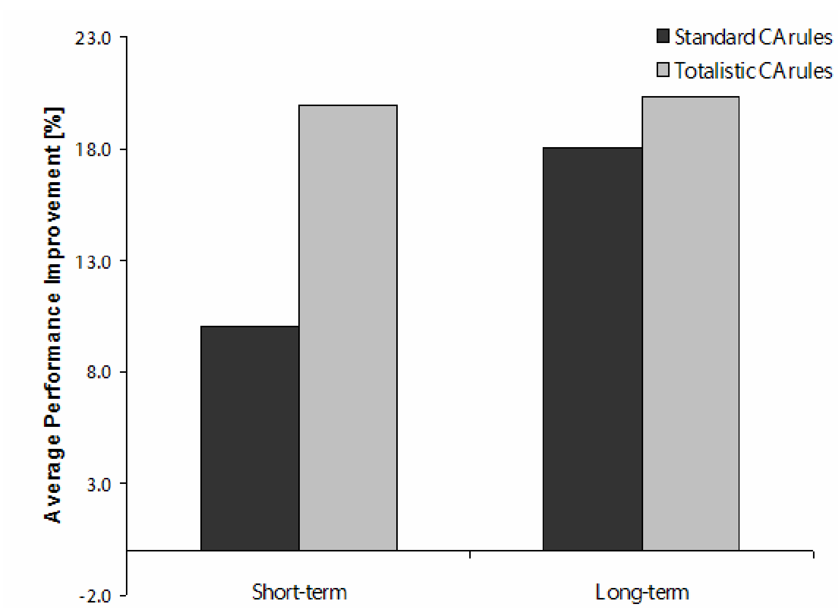


Figure 134. Comparison of the average performance improvements produced in the morphogenic evolutionary design of the entire steel structural systems with standard and totalistic CA rules in the short-term and long-term experiments

As in the previous section, I compared the results of the short-term and the long-term morphogenic evolutionary design experiments to the results obtained in the evolutionary optimization experiments. Figure 135 shows the average performance improvements between the morphogenic evolutionary design and the evolutionary optimization achieved in the conducted experiments. It clearly shows that both morphogenic evolutionary design processes significantly outperformed the evolutionary optimization processes both in the short-term and in the long-term experiments. The obtained performance improvement levels were higher than any achieved in the experiments reported in this dissertation. They exceeded 16 percent and 14 percent in the short-term and the long-term processes, respectively.

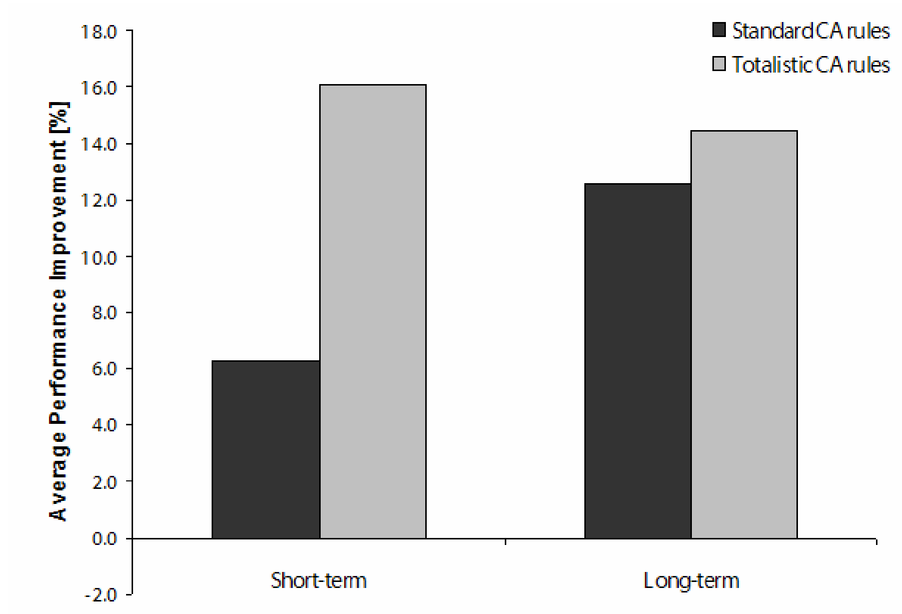


Figure 135. Comparison of the average performance improvements produced in the morphogenic evolutionary design and the evolutionary optimization of the entire steel structural system with standard and totalistic CA rules in the short-term and long-term experiments

8.4. Summary

In the design experiments reported in this chapter, I experimentally investigated the new engineering design paradigm inspired by the developmental processes occurring in nature (generative representations) and the processes of evolution (evolutionary algorithms). It was named morphogenic evolutionary design (see chapter 4).

The experimental results reported in this chapter constitute the third and last stage of the Empirical Performance Validation process (see section 3.6.3) in which I investigated the integrated components, i.e. the generative representation component and the evolutionary computation component, of Emergent Engineering Design. I have attempted to build confidence in the usefulness of the integrated components of EED by reporting and discussing the results of a large number of morphogenic evolutionary design experiments.

In the first section of this chapter, I restated criteria of novelty and optimality of steel structural systems in tall buildings which were previously defined in chapters 6 and 7). As before, I also revisited the fundamental research question and the fundamental research hypothesis and refined them in the context of design problems considered in this dissertation.

In the second section of this chapter, I reported the results of the morphogenic evolutionary design of wind bracing systems. In three subsections, I investigated morphogenic evolutionary design processes with different types of generative representations: based on elementary CAs, based on 1D CAs, and based on 2D CAs. I conducted extensive sensitivity analyses in the short-term morphogenic evolutionary design experiments to determine the optimal values of the evolutionary computations parameters and the generative representations parameters. The results showed that the optimal values found here were different than the ones identified in the evolutionary optimization experiments. They also showed that the morphogenic evolutionary design processes achieved high level of performance improvement. They produced better design concepts than design concept generators studied in chapter 6. Also, in most of the cases (with one exception only), they outperformed evolutionary optimization processes in optimizing wind bracing systems. At the same time, morphogenic evolutionary design processes generated interesting structural shaping patterns which were qualitatively different than the patterns produced in the evolutionary optimization processes.

In the third section of this chapter, I investigated morphogenic evolutionary design of the entire steel structural systems in tall buildings. I showed empirically that the integrated components of EED performed well in this complex problem domain and achieved very high levels of performance improvement which exceeded 20 percent.

In the next chapter, I will discuss the final stage of the validation process of EED, namely Theoretical Performance Validation. I will also describe the contribution of this dissertation to the field of engineering design. Finally, I will briefly discuss the limitations of the proposed approach as well as the most promising directions of future research.

9. CLOSURE

"By wisdom a house is built and through understanding it is established, through knowledge its rooms are filled with rare and beautiful treasures."

(King Solomon, Proverbs, 24:3)

In this dissertation, I proposed, presented, developed, and tested a new design method, called Emergent Engineering Design, which uses models based on complex systems and inspired by the processes occurring in nature to represent major phases of engineering design processes. The objective of this chapter is to bring the development of Emergent Engineering Design to a closure by demonstrating that I have accomplished the fundamental research objective of this dissertation and have answered the research questions posed (see chapter 3).

I will do it by first showing that new scientific knowledge has been added to the field of engineering design (see section 9.1). The Validation Square methodology will be used to demonstrate that. Next, I will describe the contributions of the research presented in this dissertation by discussing the new knowledge added to the field of engineering design (see section 9.2.1), showing its originality and significance (see section 9.2.2), and presenting research deliverables (see section 9.2.3).

Furthermore, I will discuss the limitations of Emergent Engineering Design (see section 9.3) and suggest most promising directions of the future research (see section 9.4). Finally, in section 9.5, I will provide concluding remarks.

9.1. Research Validity

In this section, I will again use the Validation Square methodology to show that Emergent Engineering Design adds new knowledge to the field of engineering design. First, in section 9.1.1, I will revisit the research questions and the research hypotheses which were posed in section 3.3. Next, in section 9.1.2, I will provide an overview of the procedure for validating Emergent Engineering Design. In sections 0 through 9.1.6, I will report the results of the four stages of the validation process (described in the previous chapters of this dissertation) and demonstrate that new knowledge has been added to the field of engineering design.

9.1.1. Revisiting the Research Questions and Hypotheses

The validation of new scientific knowledge in the context of Ph.D. research rests on three major elements (Pedersen 1999):

1. Answering the posed research questions
2. Conformity of the answers with the research hypotheses
3. Acceptability of the answers from the Ph.D. requirement perspective

Hence, I will begin with revisiting the fundamental research question and the fundamental research hypotheses of this dissertation.

Fundamental Research Question
How can one construct an effective method for designing engineering systems that would support development of novel designs and their efficient optimization?
Fundamental Research Hypothesis
Emergent Engineering Design, a design method in which all major elements of engineering design (i.e. design representation, actual design process, and design evaluation) are modeled as complex systems, can effectively produce novel designs and efficiently optimize them.

As discussed in section 3.3, the fundamental research question has been divided into 4 research questions in order to facilitate the development of the proposed method in a more structured way. The research questions and the corresponding hypotheses are presented below.

Research Question 1 (Represent):
Based on the existing knowledge on how to represent engineering systems; what mechanisms and models can be used to produce novel designs?
Research Hypothesis 1 (Represent):
Evolutionary design and complex systems provide a framework for defining generative representations, i.e. representations of engineering systems based on simple programs, which can successfully produce novel designs.
Research Question 2 (Decompose):
Knowing that complex engineering design problems are usually decomposed into sub-problems; how can a decomposition of an engineering system be defined and how can a decomposed system be effectively designed?
Research Hypothesis 2 (Decompose):
Cooperative coevolutionary models provide an efficient framework for a decomposition of complex design problems and conducting design processes using cooperative coevolutionary algorithms.
Research Question 3 (Generate and Optimize):
One of the major objectives of almost all engineering design processes is achieving optimality; what mechanisms should be used to efficiently optimize engineering designs?
Research Hypothesis 3 (Generate and Optimize):
Evolutionary computation provides a framework for conducting engineering design processes and optimizing engineering designs.
Research Question 4 (Evaluate):
Evaluation of design concepts is one of the most important stages of a design process; how can the evaluation process be performed to accomplish robustness of designs?
Research Hypothesis 4 (Evaluate):
Competitive coevolutionary models are suitable for adaptive testing and evaluation of engineering design concepts and can successfully increase robustness of generated designs.

The relationship of the phases of Emergent Engineering Design to the four research questions and hypotheses is presented in Figure 136.

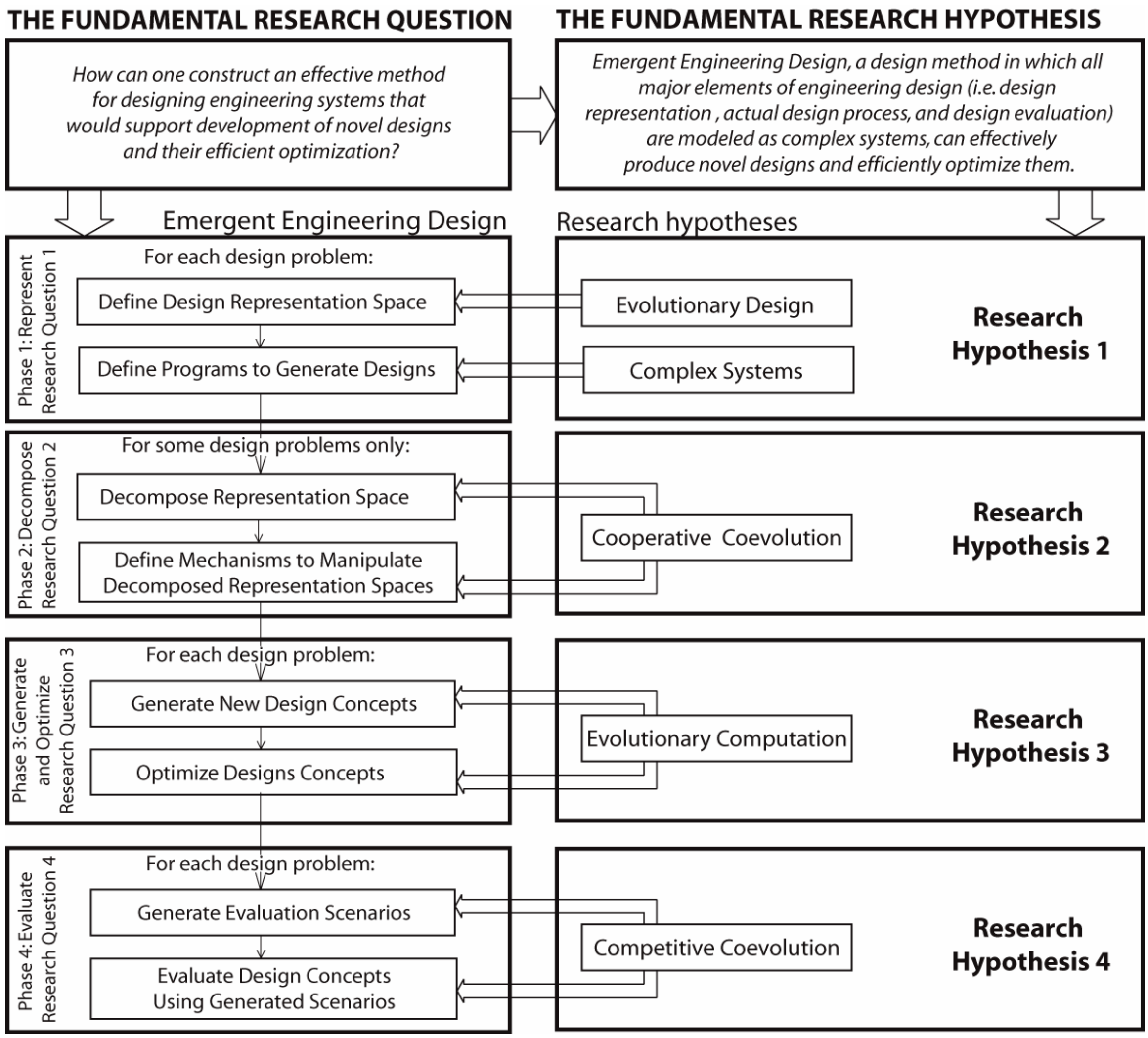
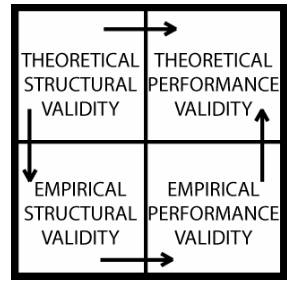


Figure 136. Phases of Emergent Engineering Design and their relationship to the four research questions and hypotheses

In this dissertation, I specifically addressed the research questions No.1 and No.3. The research questions No.2 and 4 will become part of the future work, as discussed in section 9.3. As stated earlier, the validation of the research in the context Ph.D. requirements is based on answering the research questions according to the hypotheses in a satisfactory manner. In this dissertation, the answers correspond to the research hypotheses and the hypotheses were tested for validity according to the process described in section 3.6. An overview of the validation process and its relationship to the chapters of this dissertation is presented in the following section.

9.1.2. Validating New Scientific Knowledge – an Overview

The process of answering the research questions posed is directly related to the process of validating the corresponding hypotheses. Hence, in order to answer the fundamental research question, each of the supporting hypotheses, i.e. the hypotheses No.1 and 3, had to be validated. The validation methodology used in this dissertation and based on the Validation Square was discussed in section 3.6. The four quadrands of the square shown to the right represent four steps of the validation process, i.e. Theoretical Structural Validity, Empirical Structural Validity, Empirical Performance Validity, and Theoretical Performance Validity. Each of these steps was applied to test the supporting hypotheses. The following subsections summarize the results of the validation process of EED. Figure 137 shows the relationships between the chapters of this dissertation and places where the results of each of the steps of the validation process were reported.



9.1.3. Testing the Theoretical Structural Validity

Theoretical Structural Validity (TSV) was the first stage of the validation process. As discussed in section 2.5.2, it is based on 'correctness' of the individual components constituting the design method and the internal consistency of the way the components are integrated in the method. By accepting TSV we can assert that the results produced by the design method are obtained in a correct and consistent manner, i.e. for a valid input the method produces a valid output.

The 'correctness' of the individual components of Emergent Engineering Design was demonstrated by providing extensive literature references while the internal consistency of EED was shown by demonstrating how the components were integrated together and presenting flow-chart representations of the method.

Specifically, TSV of the individual components of EED, i.e. the generative representation components and the evolutionary computation component, was demonstrated in the following way:

- TSV of the generative representation component was demonstrated in sections 2.1.3 (Design Representations), 2.2 (Cellular Automata), and 2.3 (Complex Systems).
- TSV of the evolutionary computation component was demonstrated in section 2.1 in general and in section 2.1.7 in the specific context of structural engineering applications.

  The internal consistency of EED was demonstrated in chapters 4 and 5. Chapter 4 showed how the generative representation and evolutionary computation components are integrated. Chapter 5 described in detail all phases of the design method and illustrated them with separate flow-charts emphasizing the information flow among and within its components.

Thus, by taking TSV of the hypothesized components of EED and the internal consistency of the proposed method, I can assert that Emergent Engineering Design is **Theoretical Structural Valid**. √

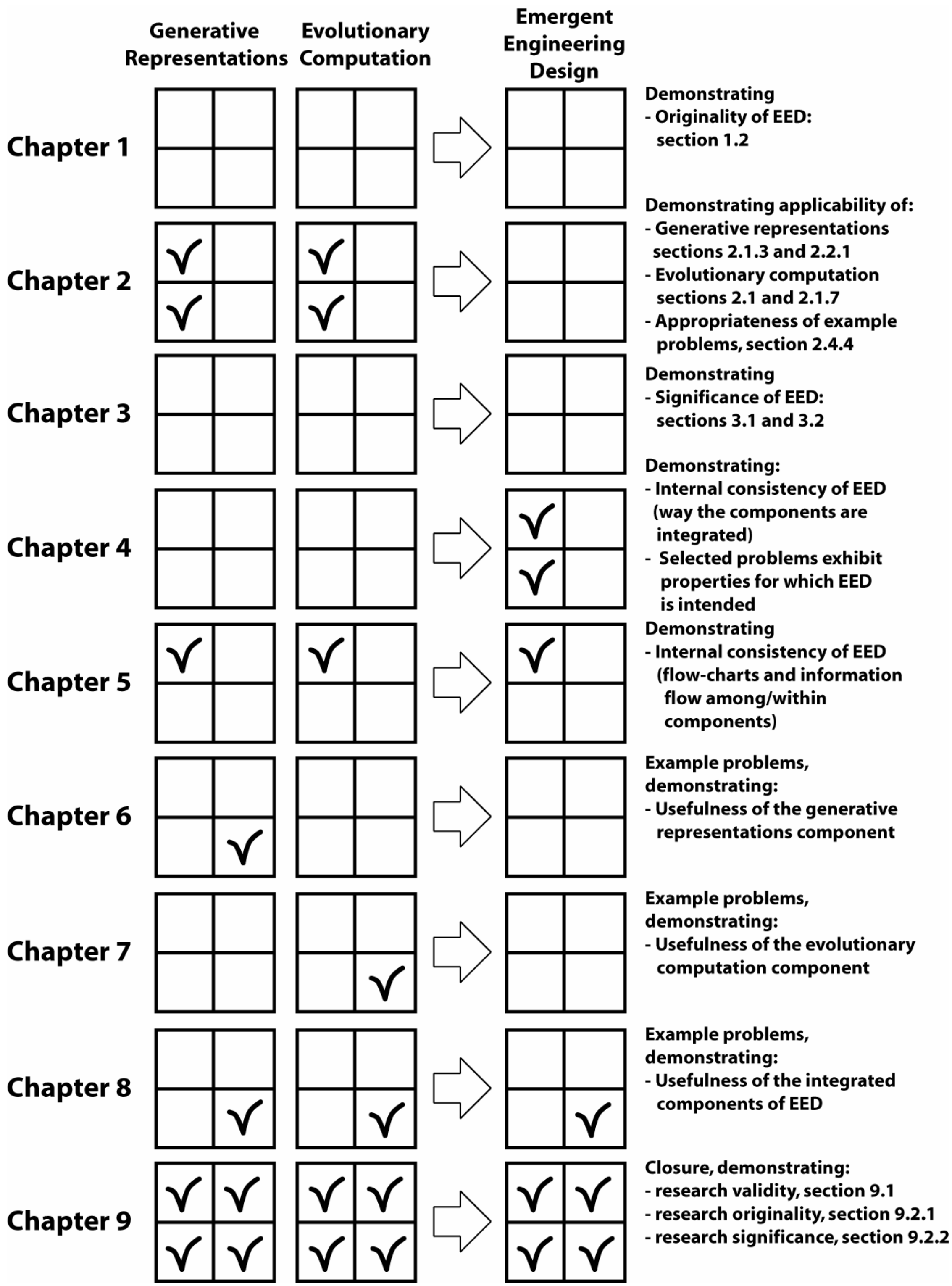


Figure 137. Overview of the relationships between the chapters of this dissertations and places where the hypotheses were tested

9.1.4. Testing the Empirical Structural Validity

Empirical Structural Validity (ESV) formed the second stage of the validation process. As discussed in section 2.5.2, ESV is based on accepting the appropriateness of the example problems that are used to verify the performance of the method.

This was done in the following way:

- By demonstrating that the example problems are similar to the problems for which EED components are generally accepted.
  This was achieved by providing state-of-the-art overviews of all components of the proposed design method in chapter 2. The overviews discussed current research developments in these fields from the perspective of their relevance to engineering design. Moreover, each section in chapter 2 contained a subsection presenting structural engineering applications, if any, of the main ideas discussed there.
- By showing that the example problems represent the actual problems for which EED is intended.
  The justification for the choice of the two example problems was presented in section 2.4.4. Also, chapter 4 demonstrated that the selected problems exhibit the properties of problems for which EED is intended.
- Documenting that the data associated with the example problems can support a conclusion.
  As discussed in section 2.4.4, the example problems investigated in this dissertation were considered as one of the most complex and time-consuming design tasks in structural engineering. Therefore, they were of suitable complexity for the demonstration of the usefulness of the proposed design method.

Given that the example problems of conceptual design of wind bracing systems and conceptual design of the entire steel structural systems are appropriate for testing Emergent Engineering Design, I assert that EED is **Empirical Structural Valid**. √

9.1.5. Testing the Empirical Performance Validity

Empirical Performance Validity (EPV) formed the third stage of the validation process. EPV is based on accepting that EED produces useful results for the selected example problems and that the components of EED contribute positively to this usefulness. EPV was demonstrated in the following way:

- The usefulness of the generative representations component in generating novel design concepts of wind bracing systems and the entire steel structural system in tall buildings was demonstrated in chapter 6.
- The usefulness of the evolutionary computation component in optimizing design concepts of wind bracing systems and the entire steel structural system in tall buildings was demonstrated in chapter 7.
- The usefulness of the integrated components of EED in producing novel design concepts and efficiently optimizing them was demonstrated in chapter 8.

Having demonstrated that Emergent Engineering Design is useful in producing novel design concepts and efficiently optimizing them, and that the components of the EED positively contribute to this usefulness, I can assert that EED is useful at least for the example problems. Hence, I can assert that EED is **Empirical Performance Valid**. √

9.1.6. Testing the Theoretical Performance Validity

Theoretical Performance Validity (TPV) is the last stage of the validation process. It is based on building confidence that EED is valid beyond the example problems, i.e. it is more general. TPV of EED will be demonstrated as follows.

The confidence that Emergent Engineering Design can produce useful results beyond the example problems, i.e. the conceptual design of wind bracing systems and the conceptual design of the entire steel structural systems in tall buildings, will be build based on the results of the previous validation steps, namely:

- TSV of EED demonstrated that the components of EED are applicable beyond the example problems (see sections 2.1 and 2.2).
- ESV showed the selected example problems represent a more general class of problems for which EED is intended.
- EPV demonstrated that EED was at least useful for the selected example problems.

The final acceptance that EED is **Theoretical Performance Valid** requires, however, a 'leap of faith' (Pedersen et al. 2000). √

TPV concludes the validation of the hypotheses of this dissertation and hence, the answering of the research questions posed. Thus, it can be accepted that new scientific knowledge has been added to the field of engineering design. I can also assert that I have accomplished the ultimate objective of this dissertation, i.e. I have *developed an engineering design method based on models of complex systems that provides a conceptually coherent framework for producing novel designs and their efficient optimization*. √

The originality and significance of Emergent Engineering Design will be discussed in the following section.

9.2. Contributions

This section outlines the contributions this dissertation makes to the field of engineering design. The contributions from the successful development and implementation of the proposed method of engineering design can be expressed in terms of its validity, usefulness, and novelty.

9.2.1. Contributions to the Field of Engineering Design

The major contributions of Emergent Engineering Design to the field of engineering design are listed below:

- It establishes an integrated and conceptually coherent framework for engineering design based on complex systems.
- It introduces a design method that is inspired by the processes occurring in nature. All the processes involved in generation, evolution, and evaluation of engineering designs model processes occurring in nature.
- It emphasizes both important aspects of the design process, i.e. novelty and optimization.
- It proposes novel ways of representing engineering systems based on cellular automata.

A graphical representation of the contributions of Emergent Engineering Design and the foundations it is built on is shown in Figure 138.

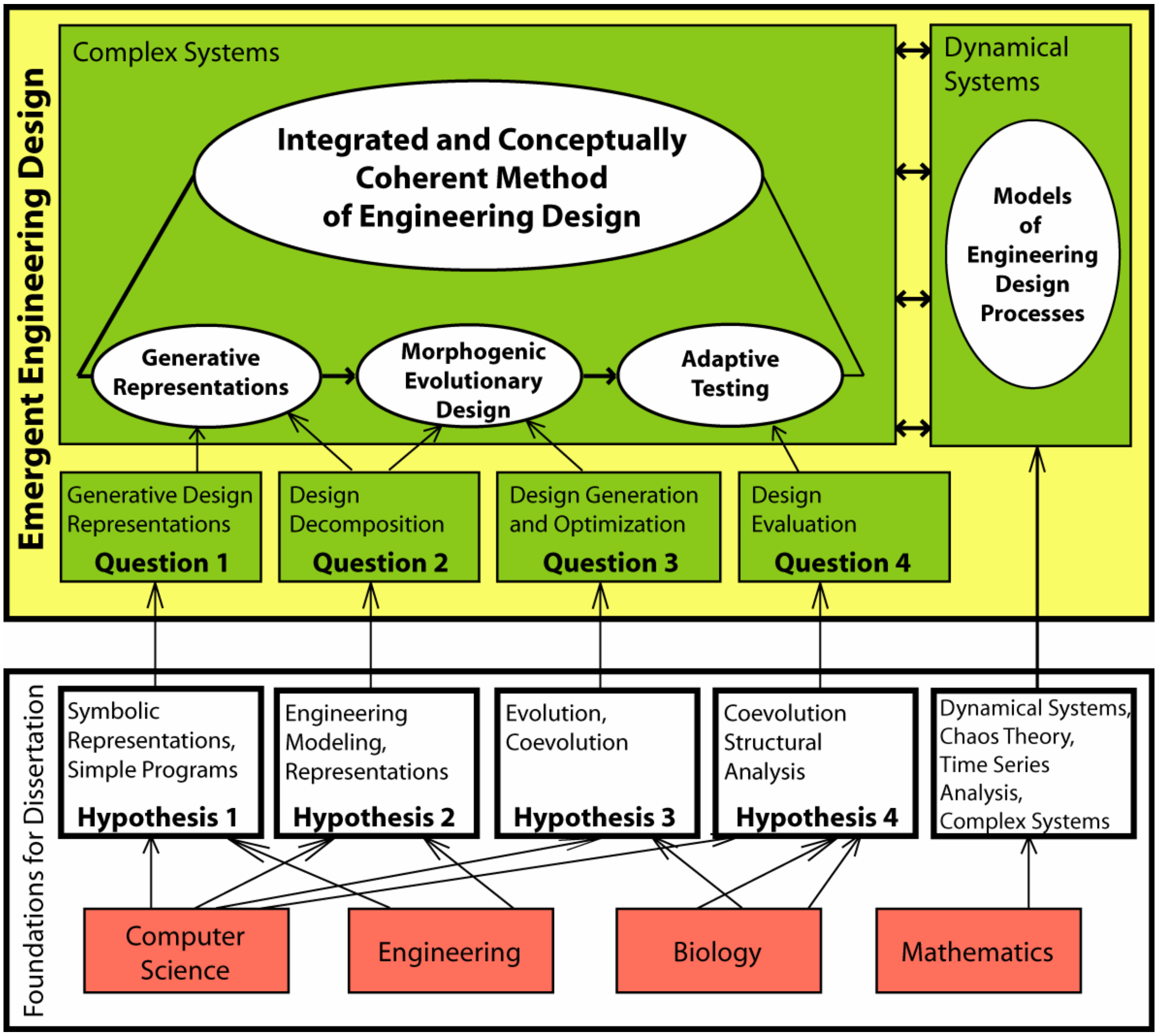


Figure 138. Contributions of Emergent Engineering Design and the foundations it is based on

9.2.2. Originality and Significance

As it was discussed in sections 2.1.3 and 2.4.2, there have been suggested many approaches to develop methods for engineering design. However, in my opinion, most of them were focused exclusively on only one of the two important aspects of engineering design, i.e. either on creativity, or on optimization. In this dissertation, I proposed, developed and implemented a design method which addresses both of these aspects.

Another issue is that many of the proposed design methods tended to be assembled from conceptually diverse components and thus not giving a coherent view of the design process. On the contrary, Emergent Engineering Design represents an integrated approach to engineering design based on models of complex systems.

Yet another significant contribution of this dissertation is a demonstration that engineering design processes can be greatly enhanced by nature. Models of processes occurring in nature can be successfully used to represent engineering systems and design processes. Thus, this dissertation builds a bridge between design by nature and engineering design.

The generality of models, procedures, and algorithms proposed in this dissertation makes them well-suited for a wide range of engineering design applications. Some of the potential applications of Emergent Engineering Design are discussed in the next section.

9.2.3. Research Deliverables

The ultimate dissertation objective was stated in section 3.3. Hence, the design method based on models of complex systems is the major deliverable of this research. The method has been implemented in a design support tool called Emergent Designer. Thus, apart from the major deliverable, the dissertation produced the following results that can be divided into four groups:

1. A consistent system of models, procedures, and algorithms regarding engineering design with a strong emphasis on both novelty and optimization.
2. A class of representations based on models of complex systems of both engineering systems and design processes inspired by the processes occurring in nature.
3. Emergent Designer, a design support tool implementing the proposed design method.
4. Experimental results in the domain of steel structural systems in tall buildings that proved feasibility, novelty, and potential practical value of the proposed design method.

9.3. Limitations

Extensive empirical studies conducted in this dissertation allowed me to identify several limitations of the proposed design method and of the generative representations proposed in this dissertation. They are presented below:

1. The sizes of cellular automata rules spaces grow rapidly when the number of cell values increases (e.g. in two-dimensional CAs). Thus, the representations proposed in this dissertation, particularly the ones based on standard CA rules, might not be satisfactory for design problems in which attributes have many possible values, e.g. more then 10. For these types of problems, more sophisticated representations may be necessary (see the discussion in the next section describing the future work).
2. The proposed generative representations based on elementary and one-dimensional CAs effectively reduce the sizes of the design spaces and bias the search process towards generation of design concepts exhibiting interesting patterns. In some cases, however, this reduction of the search space may be disadvantageous and optimal design concepts may be lost because they simply cannot be generated by these representations (see for example the results of morphogenic evolutionary design of wind bracing systems composed of simple X bracings presented in section 8.2.1).

9.4. Future Work

The work presented in this dissertation can be extended in many ways. First, the remaining two phases of the engineering design process, i.e. design decomposition and design evaluation, will be studied and answers to the research questions presented in section 3.3 will be sought.

Another potential area of further development of Emergent Engineering Design is in the exploration of various types of simple programs, including L-systems, mobile automata, and more elaborate versions of cellular automata (e.g. non-uniform CAs, continuous CAs), to model engineering systems and design processes.

Finally, as mentioned in the previous section, the generality of this method makes it well-suited for a wide range of engineering design problems. Thus, Emergent Engineering Design will be applied to other types of discrete design problems, e.g. design of space structures, or design of bridges. It offers also a great potential for continuum design problems, e.g. topology optimization of plates or shell structures. In this case, it can be easily combined with traditionally used design evaluation methods based on finite element analysis.

9.5. Concluding Remarks

It is my hope that the work on Emergent Engineering Design will be continued and will provide some inspiration as well as new questions and problems for other researchers working in this field. I believe that nature inspired methods offer enormous potential for many fields of engineering and other disciplines. They may introduce a new paradigm of engineering design in which nature's potential for novelty and optimality will be successfully used to better design the world we inhabit.

APPENDIX A

In this appendix, a chronological overview of the major applications of evolutionary computation in structural design is presented. The overview begins with the initial applications of evolutionary algorithms in sizing optimization of simple truss systems in the mid 1980's and provides a summary of major developments in this area until now. A detailed discussion of the field and its most promising future research directions was presented earlier in section 2.1.7.

The applications are classified with respect to several criteria, including:

- application domain,
- representation type,
- evolutionary algorithm,
- fitness function, and
- methods of handling constraints.

A detailed discussion on the importance of these criteria for structural design applications was provided in chapter 2.

Reference	Domain	Problem	Represen-tation	EA used	Fitness function	Con-straint-handling method
(Hoeffler et al. 1973)	Shape optimization	Location of joints in truss systems	Fixed-length, real-valued vectors	ES	Single objective, weight minimization	N/A
(Lawo and Thierauf 1982)	Sizing optimization	Planar frame under earthquake loading	Fixed-length, real-valued vectors	ES	Single objective, weight minimization	N/A
(Goldberg and Samtani 1986)	Sizing optimization	Cross-sections in planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Hajela 1990)	Sizing optimization	Cross-sections in planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Sandgren et al. 1990)	Continuum TOD	Planar cantilever plates	Fixed-length, 2D binary arrays (bitarrays)	GA	Single objective, weight minimization	N/A
(Deb 1991)	Sizing optimization	Welded beams	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Jenkins 1991b) and (Jenkins 1991a)	Continuum SO	Shape of structural members	Fixed-length, 2D binary arrays (bitarrays)	GA	Single objective, weight minimization	Penalty function
(Shankar and Hajela 1991)	Discrete TOD	Planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Hajela 1992)	Sizing optimization	Cross-sections in planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A

(Hajela and Lin 1992)	Sizing optimization	Planar truss systems	Fixed-length, binary strings	GA	Multiobjective, min-max	N/A
(Jensen 1992)	Continuum TOD	Planar cantilever plates	Fixed-length, 2D binary arrays (bitarrays)	GA	Single objective, weight minimization	N/A
(Rajeev and Krishnamoorthy 1992)	Sizing optimization	Cross-sections in planar trusses	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Richards and Sheppard 1992)	Continuum SO	Shape of structural members	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Sandgren and Jensen 1992)	Continuum TOD	Planar cantilever plates	Fixed-length, 2D binary arrays (bitarrays)	GA	Single objective, weight minimization	N/A
(Adeli and Cheng 1993)	Sizing optimization	Spatial truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Chapman et al. 1993)	Continuum TOD	Planar cantilever plates	Fixed-length, 2D binary arrays (bitarrays)	GA	Single objective, weight minimization	N/A
(Lin and Hajela 1993)	Sizing optimization	Cross-sections in planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Hajela et al. 1993)	Discrete TOD	Planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Grierson and Pak 1993a)	Discrete TOD, SO, and sizing optimization	Planar frame systems	Fixed-length binary strings	GA	Single objective, weight minimization	N/A
(Schoenauer and Xanthakis 1993)	Sizing optimization	Planar truss systems	Fixed-length, real valued vectors	GA	Single objective, weight minimization	Behavioral memory
(Watabe and Okino 1993)	Continuum SO	Shape of structural members	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Sakamoto and Oda 1993)	Discrete TOD and sizing optimization	Planar truss systems	Fixed-length, binary strings	GA + optimality crite-ria me-thod	Single objective, weight minimization	N/A
(Adeli and Cheng 1994)	Sizing optimization	Spatial truss systems	Fixed-length binary strings	GA	Single objective, weight minimization	Penalty function and augmented Lagrangian
(Chapman et al. 1994)	Continuum TOD and SO	Planar cantilever plate	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Coello Coello et al. 1994)	Sizing optimization	Planar and spatial truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Koumousis and Georgiou 1994)	Discrete TOD and SO	Planar steel truss roofs	Fixed-length, binary strings	GA + logic program	Single objective, weight minimization	N/A

(Keane 1994)	Discrete SO	Planar truss system (satellite boom)	Fixed-length, binary strings	GA	Single objective, minimization of vibration	Penalty function
(Bohnenberger et al. 1995)	Discrete TOD	Pylon structures	Fixed-length, binary strings	GA and ES	Single objective, weight minimization	N/A
(Rajan 1995)	Discrete TOD , SO and sizing optimization	Spatial truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Hajela and Lee 1995a)	Discrete TOD	Planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Ohsaki 1995)	Discrete TOD	Planar truss systems	Fixed-length, binary strings	GA	Single objective, total cost	Penalty function
(Hajela and Lee 1995b) and (Hajela and Lee 1996)	Discrete TOD	Planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Immune network
(Keane and Brown 1996)	Discrete SO	Spatial truss systems (satellite boom)	Fixed-length, binary strings	GA	Single objective, minimization of vibration	N/A
(Soh and Yang 1996)	Discrete SO	Planar and spatial truss systems	Fixed-length, binary strings	GA + fuzzy logic	Single objective, weight minimization	Fuzzy logic
(Ramasamy and Rajasekaran 1996)	Discrete TOD and sizing optimization	Planar truss systems	Fixed-length, binary strings	GA + neural net-work	Single objective, weight minimization	Penalty function
(Nakanishi and Nakagiri 1996) and (Nakanishi and Nakagiri 1997)	Discrete TOD	Planar frame and panel structures	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Cheng and Li 1997)	Sizing optimization	Planar truss systems	Fixed-length, binary strings	Pareto GA, MOGA	Multiobjective, with 2 or 3 objectives	Fuzzy penalty function
(Parmee et al. 1997) and subsequent papers	Discrete TOD, SO, and sizing optimization	Various problems considered	Various encodings (binary, real, etc.)	(GA, CHC, and ES)	Single and multiobjective approaches	Various constraint-handling methods
(Yang and Soh 1997)	Discrete SO	Planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Rajeev and Krishnamoorthy 1997)	Discrete TOD, SO, and sizing optimization	Planar truss structures	Variable-length, binary strings	GA	Single objective, weight minimization	N/A
(Jenkins 1997)	Discrete SO and sizing optimization	Planar multistory frame structure with truss-supported hangers	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function

(de Barros Leite and Topping 1998)	Discrete TOD and sizing optimization	Welded beam, planar truss systems, and prestressed I-sections	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Camp et al. 1998)	Sizing optimization	Planar truss and frame structures	Fixed-length, binary strings	GA	Single objective, various fitness functions	Penalty function
(Chen and Rajan 1998)	Discrete TOD, SO, and sizing optimization	Planar frame systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Adaptive penalty function
(Nair et al. 1998)	Sizing optimization	Planar truss system	N/A	GA + approxi-mation model	Single objective, weight minimization	Penalty function
(Ohmori and Kito 1998)	Discrete TOD	Planar and spatial truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Hajela et al. 1998)	Discrete TOD	Planar and spatial grillage structures	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Soh and Yang 1998)	Discrete TOD, SO, and sizing optimization	Planar bridge trusses	Fixed-length, binary strings	GA + domain know-ledge	Single objective, weight minimization	Penalty function
(Shrestha and Ghaboussi 1998)	Discrete TOD	Planar truss systems	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Topping and de Barros Leite 1998)	Sizing optimization	Cable-stayed bridge	Fixed-length, binary strings	Parallel GA	Single objective	N/A
(Wibowo and Besari 1998)	Continuum SO	Oval axially symmetric shells	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Kita and Tanie 1998) and (Kita and Tanie 1999)	Continuum SO	Planar structures	Fixed-length, binary strings	GA	Single objective, weight minimization	N/A
(Annicchiarico and Cerrolaza 1999)	Continuum SO	Planar structures	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty function
(Hajela and Kim 1999)	Continuum structural elasticity analysis	Planar structures	Cellular automata	GA	Single objective, strain energy minimization	Penalty function
(Coello Coello and Christiansen 2000)	Sizing optimization	Planar and spatial truss systems	Fixed-length, binary strings	GA + min-max strategy	Multiobjective, weight, displacement, and stress minimization	Penalty function (death penalty)
(Pezeshk et al. 2000)	Sizing optimization	Planar multi-story frame systems	Fixed-length, binary strings	GA	Single objective, weight minimization	Penalty functions

(Arciszewski and De Jong 2001)	Discrete TOD and sizing optimization	Steel skeleton structures in tall buildings	Fixed-length, integer encodings	Parallel EA, island model	Single objective, weight minimization	Penalty function (death penalty), repair mechanisms
(Woon et al. 2001)	Continuum SO	2D spanner head and flange webbing	Fixed-length, binary strings	GA	Single objective, weight and deflection minimization	None
(Greiner et al. 2001)	Sizing optimization	Planar frame structures	Fixed length, binary strings	GA, CHC, and NSG	Single objective (weight), and multiobjective (weight and number of member cross-sections)	Penalty functions
(Murawski et al. 2001)	Discrete TOD and sizing optimization	Steel skeleton structures in tall buildings	Fixed-length, integer encodings	ES	Single objective, weight minimization	Penalty function (death penalty), repair mechanisms
(Annicchiarico and Cerrolaza 2001)	Continuum SO	3D cantilever plate with circular hole	Fixed-length, binary strings	GA	Single objective, minimization of volume	Penalty function
(Hajela and Kim 2001)	Continuum structural elasticity analysis	Planar structures	Binary and real encodings and cellular automata	GA	Single objective, strain energy minimization	Penalty function
(Nanakorn and Meesomklin 2001)	Sizing optimization	Planar truss systems and frame	Fixed-length, binary strings	GA	Single objective, weight minimization	Adaptive penalty function
(Deb and Gulati 2001)	Discrete TOD , SO and sizing optimization	Planar and spatial truss systems	Fixed-length, real valued vectors	GA	Single objective, weight minimization	Penalty function
(Deb and Goel 2001)	Continuum SO	Planar plate structures	Fixed-length, binary strings	NSGA-II + hill climber	Multiobjective, weight and displacement minimization	
(Sarma and Adeli 2001)	Sizing optimization	Spatial multistory frame structures	Fixed-length, binary strings	Parallel GA, island model	Single and multiobjective	
(Yang and Soh 2002)	Discrete TOD	Planar truss systems	Variable-length, parse trees	GP	Single objective, weight minimization	Penalty function
(Nair and Keane 2002)	Sizing optimization	Planar truss systems	Fixed-length, binary strings	CCEA	Single objective, weight minimization	Penalty function
(Azid et al. 2002)	Discrete TOD	Planar and spatial truss systems	Fixed-length, real valued vectors	GA	Single objective, weight minimization	Penalty function

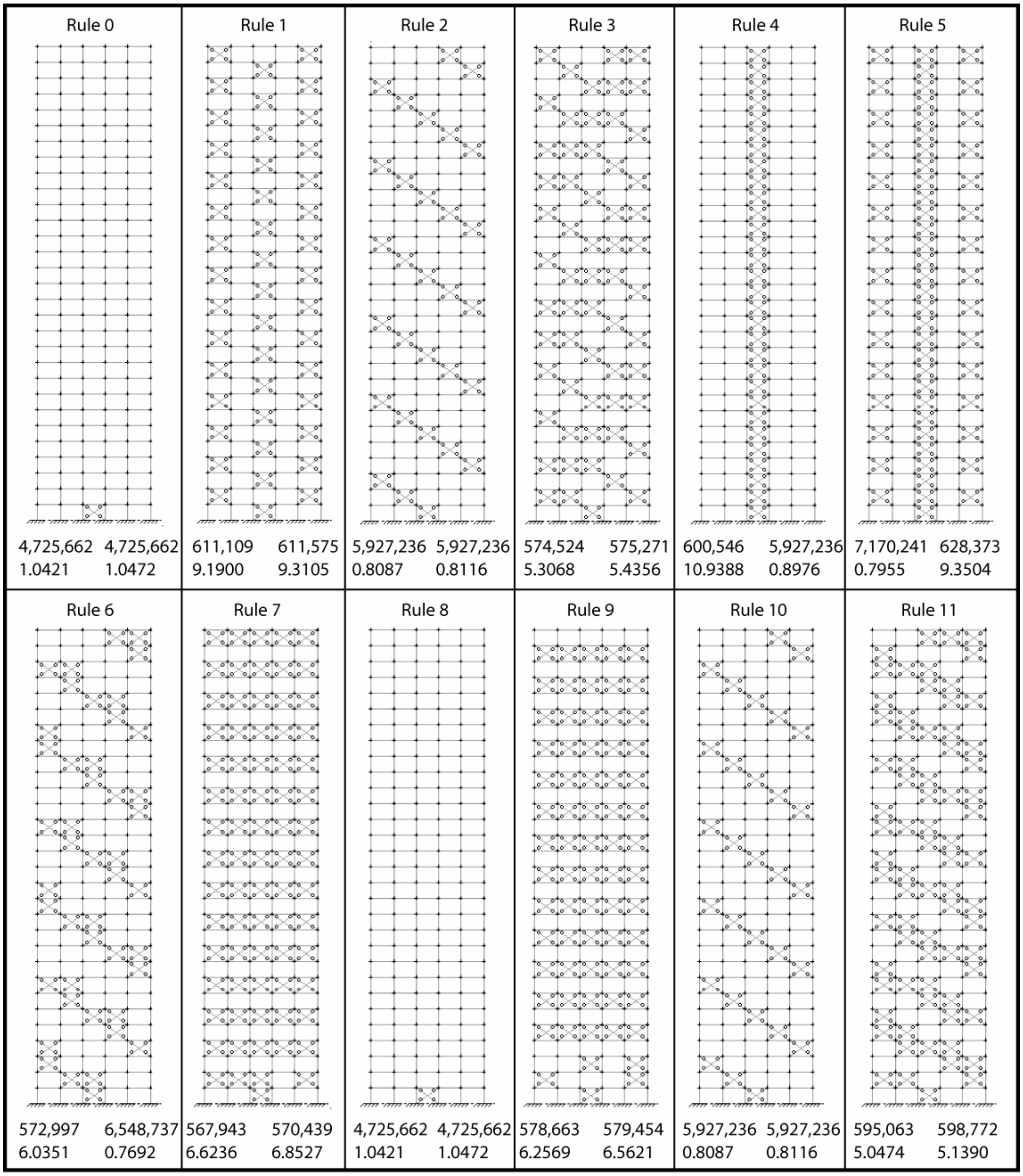
(Hamda et al. 2002a)	Continuum TOD	Planar and spatial cantilever plates	Variable-length, Voronoi-based, and fractal-based	GA	Single objective, weight minimization	Penalty function
(Hamda et al. 2002b)	Continuum TOD	Planar cantilever plate	Variable-length, Voronoi-based	NSGA-II	Multiobjective, weight and displacement minimization	
(Kicinger et al. 2003)	Discrete TOD	Steel skeleton structures in tall buildings	Fixed-length, integer representations	ES	Single objective, weight minimization	Penalty function (death penalty), repair mechanisms
(Dimou and Koumousis 2003)	Sizing optimization	Planar truss systems	Fixed-length, binary strings	Parallel GA	Single objective for individuals in each population – total cost	Penalty function
(Pullmann et al. 2003)	Discrete TOD	Reinforced concrete tall buildings	Fixed-length, integer strings	Unified EA and fuzzy sets	Single objective, total cost	Fuzzy logic
(Kicinger et al. 2004d)	Discrete TOD and Sizing optimization	Wind bracing systems in tall buildings	Generative representations based on cellular automata (1D and 2D)	ES	Single objective, the total weight	Penalty function (death penalty), repair mechanisms
(Kicinger et al. 2004c)	Discrete TOD and Sizing optimization	Steel structural systems in tall buildings	Generative representations based on 1D cellular automata	ES	Single objective, the total weight	Penalty function (death penalty), repair mechanisms
(Kicinger et al. 2004a)	Discrete TOD and sizing optimization	Steel structural systems in tall buildings	Fixed-length, integer representations	Distributed EA (island-model)	Single objective, the total weight	Penalty function (death penalty), repair mechanisms
(Kicinger and Arciszewski 2004)	Discrete TOD and sizing optimization	Steel structural systems in tall buildings	Fixed-length, integer representations	ES	Multiobjective (aggregate function), the total weight and the maximum horizontal displacement	Penalty function (death penalty), repair mechanisms

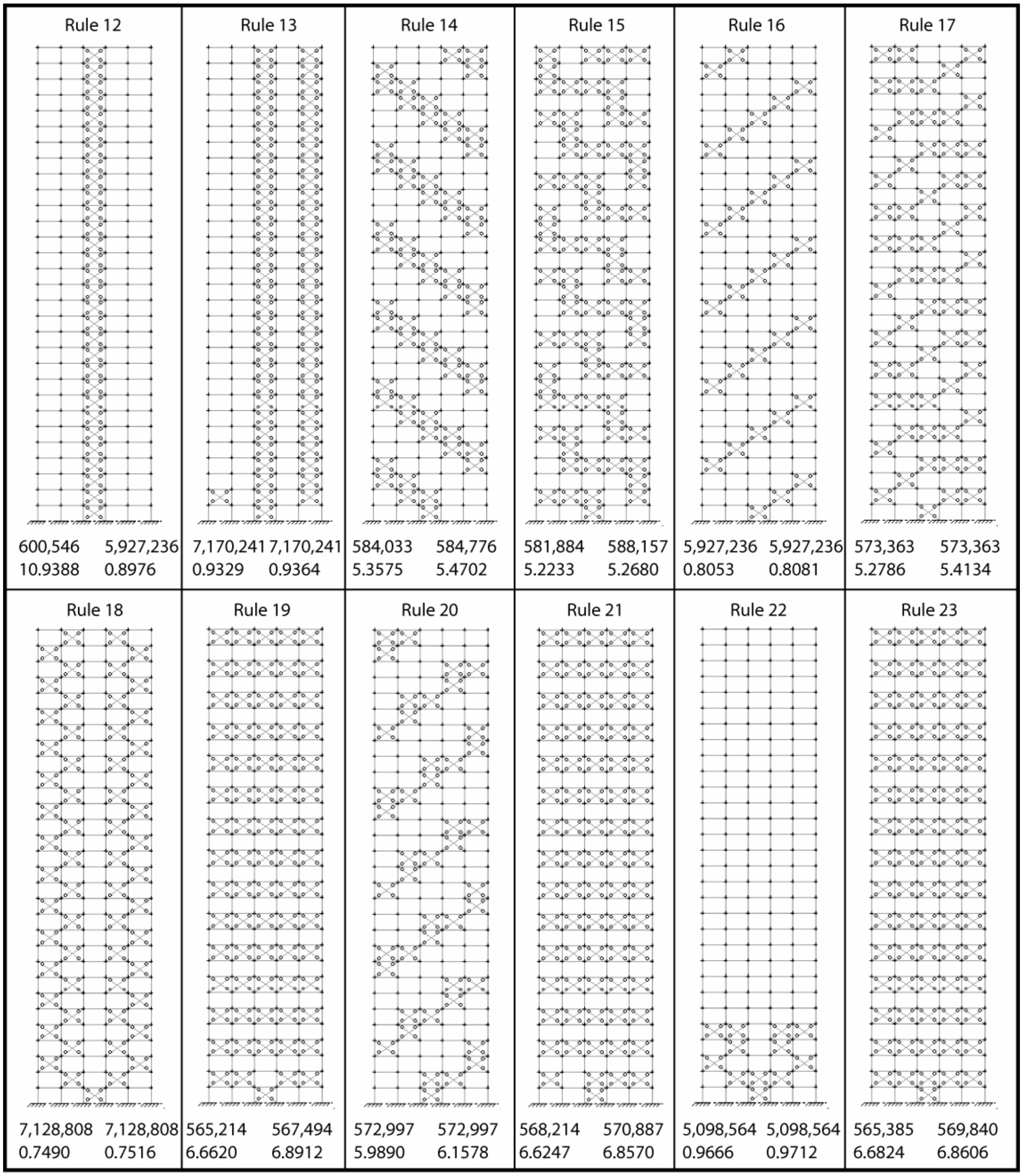
APPENDIX B

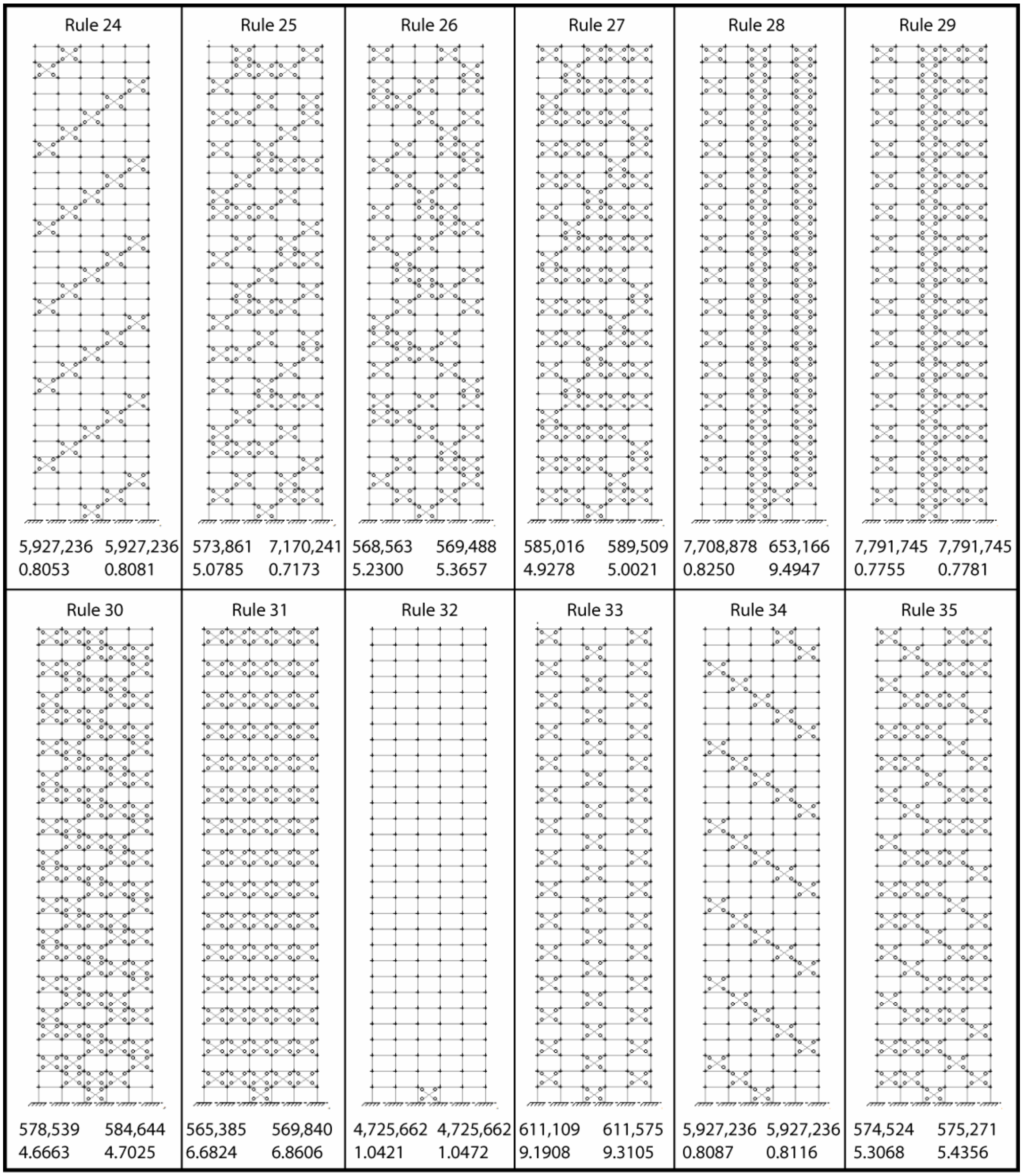
In this appendix, the entire set of 256 design concepts of wind bracing systems in tall buildings generated by elementary CAs is presented. The designs have been generated from the simplest design embryo consisting of a single X bracing located in the central bay. All designs shown in the following table were generated by elementary CAs with periodic boundary conditions.

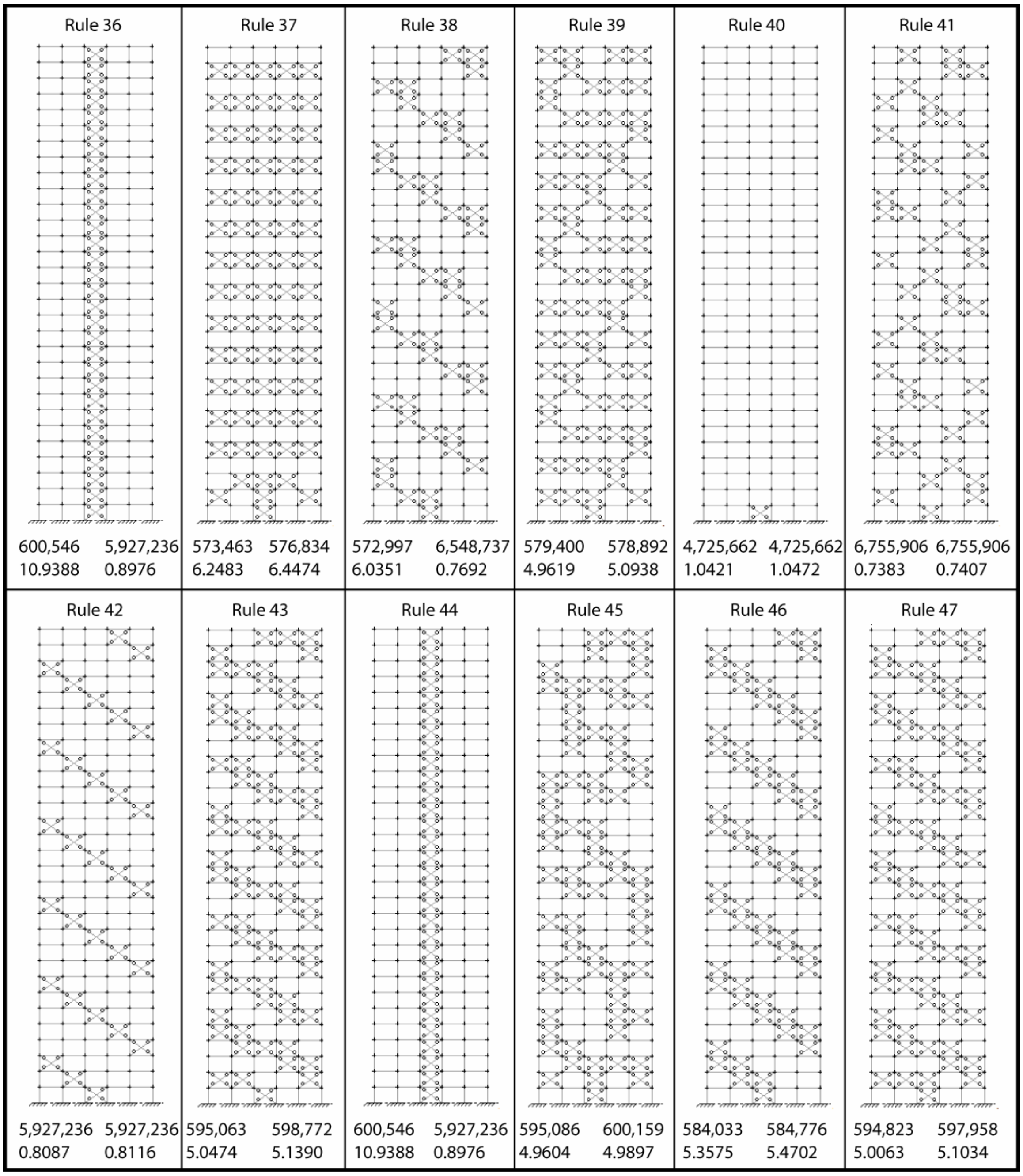
Each cell in this table contains the number of the design rule at the top, the actual design developed from the design embryo by this rule (center), and four values arranged in a 2 x 2 array (the bottom part) as shown on the right. This array contains four values representing the total weight of the steel structural system (the first row) and its maximum horizontal displacement (the second row). The first column contains measurements obtained using the first-order structural analysis while the second column contains the values produced by a more accurate and at the same time more computationally expensive P-Δ analysis. The values of the total weight of the steel structural system presented in the first row are measured in lbs. whereas the values of the maximum horizontal displacement, shown in the second row, are measured in inches.

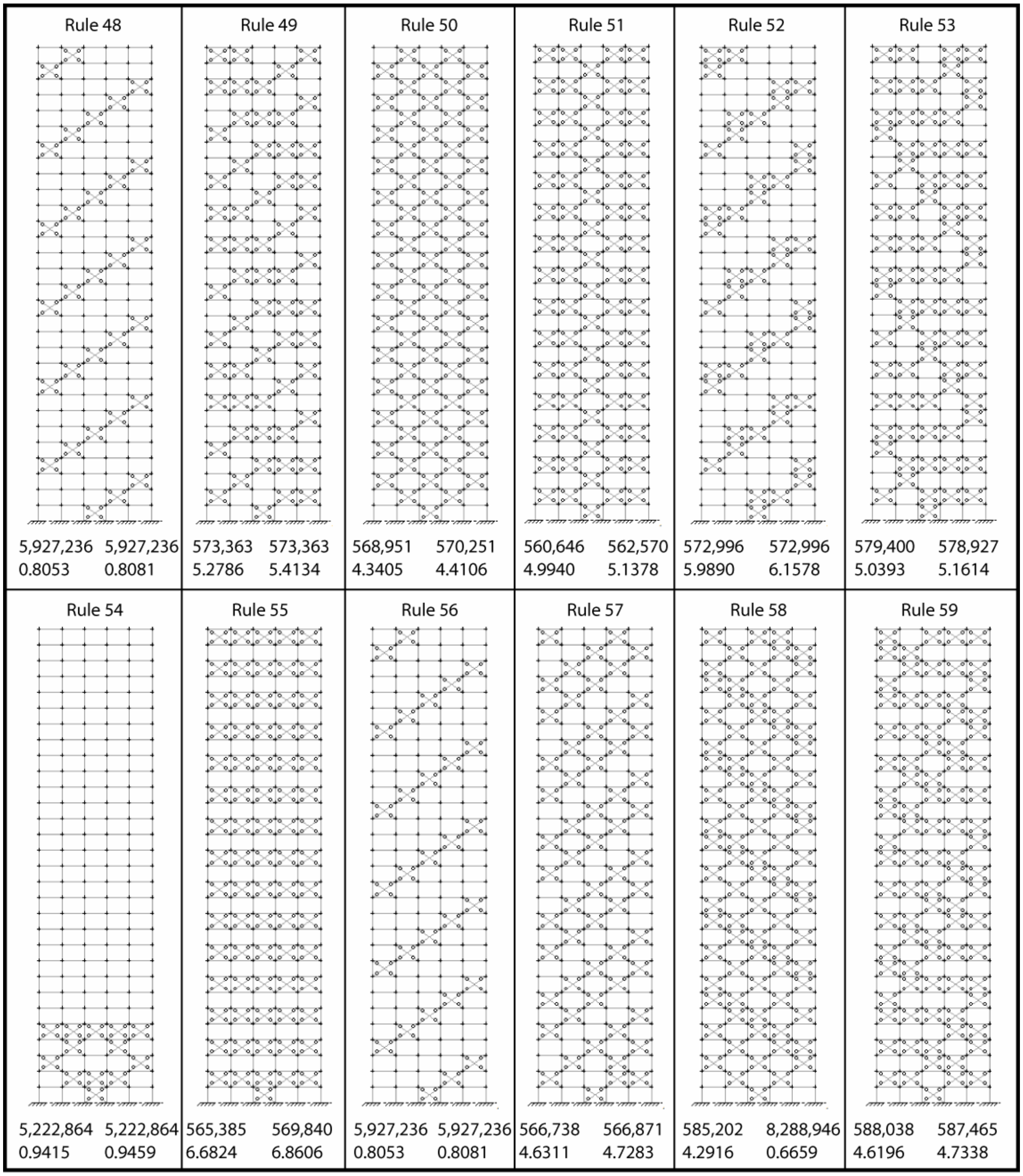
Total weight of the steel structure (1st-order)	Total weight of the steel structure (P-Delta)
Maximum horizontal displacement (1st-order)	Maximum horizontal displacement (P-Delta)

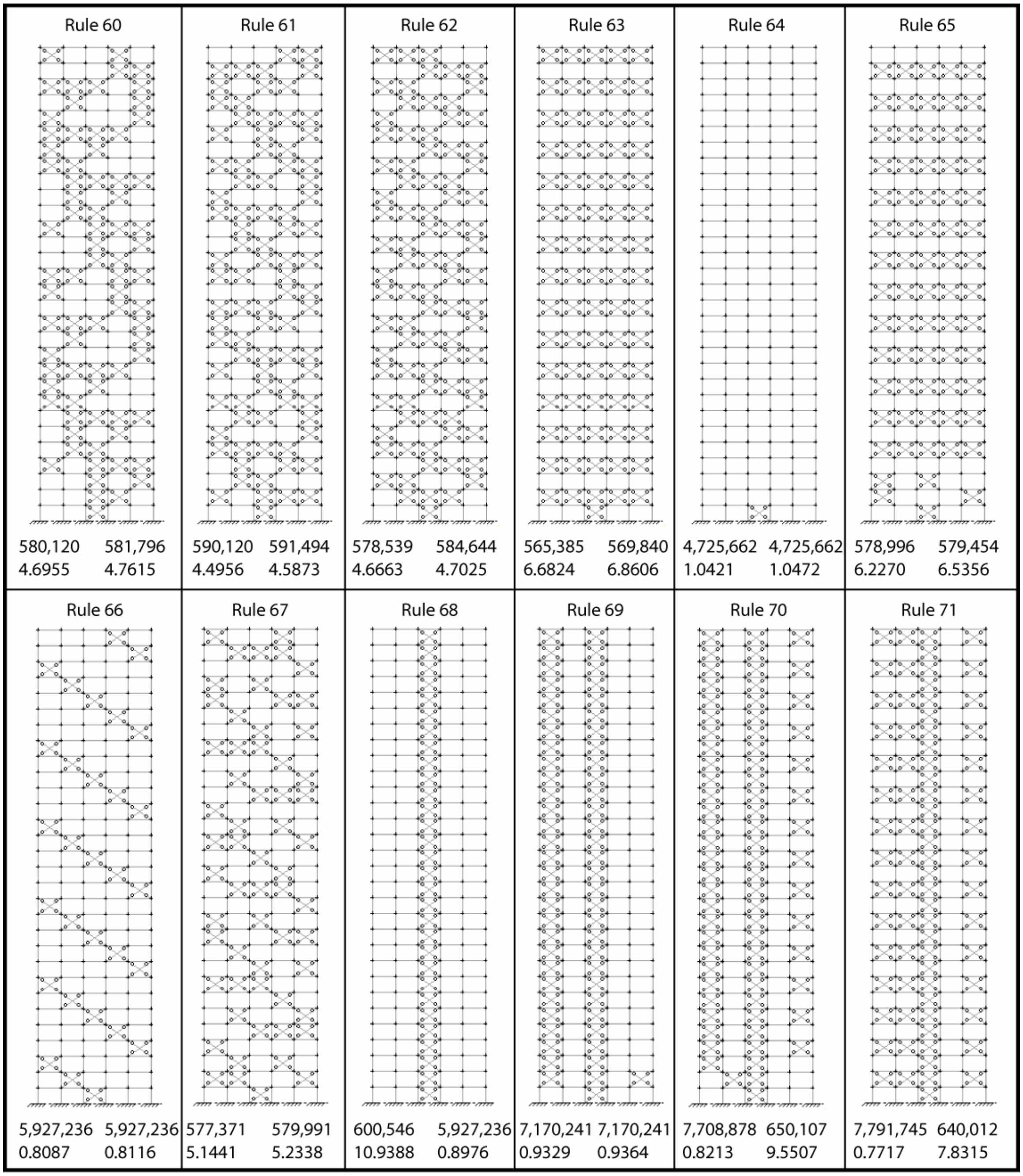

Rule 0
4,725,662 4,725,662
1.0421 1.0472
Rule 1
611,109 611,575
9.1900 9.3105
Rule 2
5,927,236 5,927,236
0.8087 0.8116
Rule 3
574,524 575,271
5.3068 5.4356
Rule 4
600,546 5,927,236
10.9388 0.8976
Rule 5
7,170,241 628,373
0.7955 9.3504
Rule 6
572,997 6,548,737
6.0351 0.7692
Rule 7
567,943 570,439
6.6236 6.8527
Rule 8
4,725,662 4,725,662
1.0421 1.0472
Rule 9
578,663 579,454
6.2569 6.5621
Rule 10
5,927,236 5,927,236
0.8087 0.8116
Rule 11
595,063 598,772
5.0474 5.1390

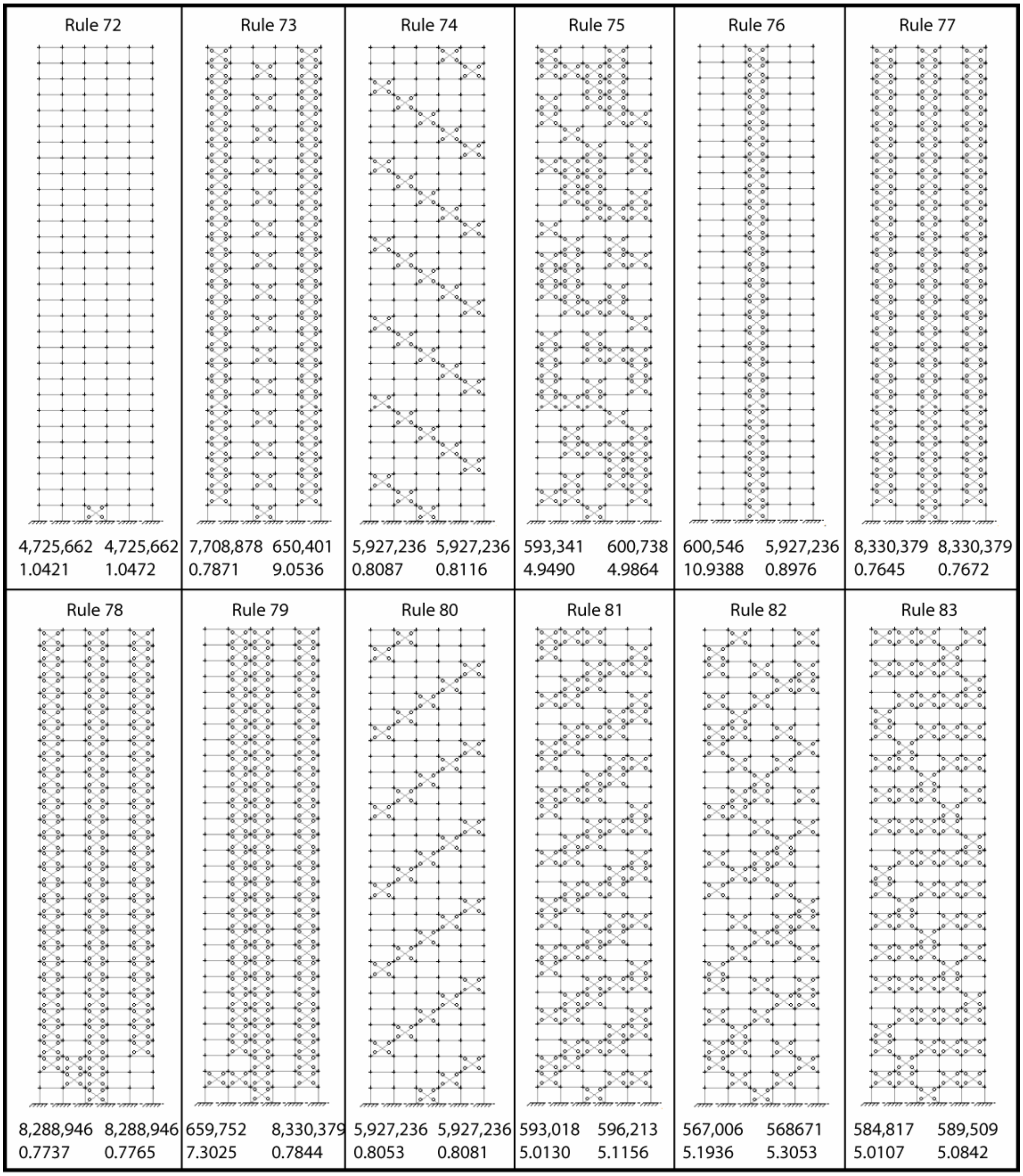

Rule 12
600,546 5,927,236
10.9388 0.8976
Rule 13
7,170,241 7,170,241
0.9329 0.9364
Rule 14
584,033 584,776
5.3575 5.4702
Rule 15
581,884 588,157
5.2233 5.2680
Rule 16
5,927,236 5,927,236
0.8053 0.8081
Rule 17
573,363 573,363
5.2786 5.4134
Rule 18
7,128,808 7,128,808
0.7490 0.7516
Rule 19
565,214 567,494
6.6620 6.8912
Rule 20
572,997 572,997
5.9890 6.1578
Rule 21
568,214 570,887
6.6247 6.8570
Rule 22
5,098,564 5,098,564
0.9666 0.9712
Rule 23
565,385 569,840
6.6824 6.8606

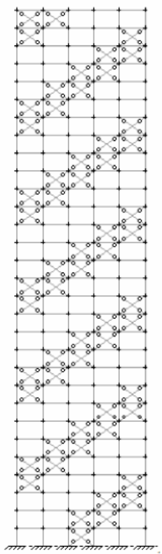
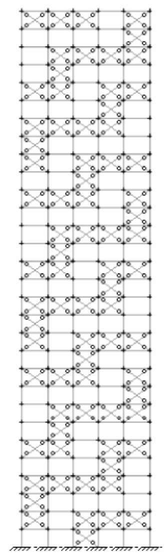
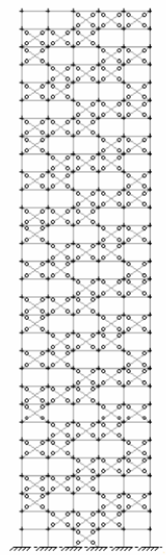
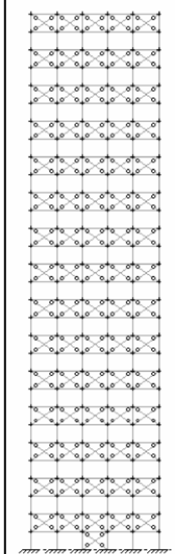
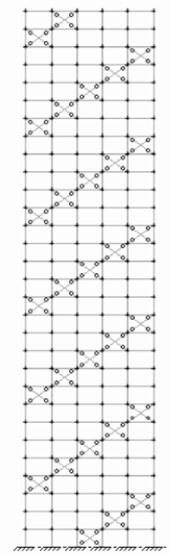
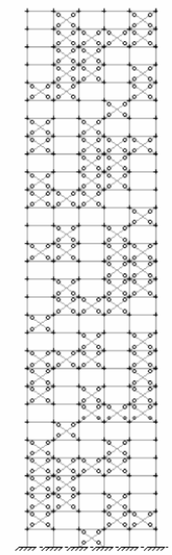
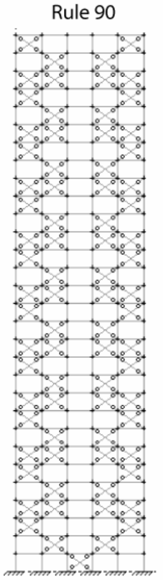
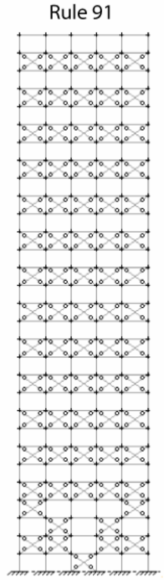
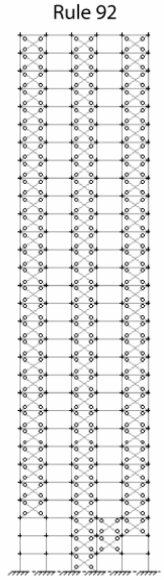
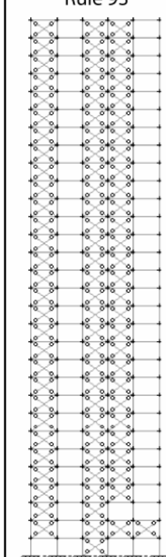
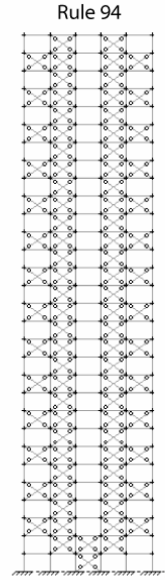
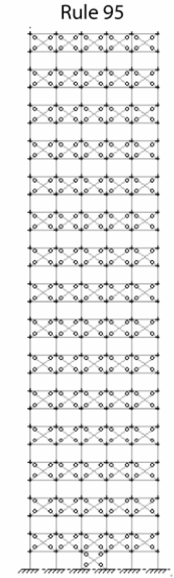

Rule 24
5,927,236 5,927,236
0.8053 0.8081
Rule 25
573,861 7,170,241
5.0785 0.7173
Rule 26
568,563 569,488
5.2300 5.3657
Rule 27
585,016 589,509
4.9278 5.0021
Rule 28
7,708,878 653,166
0.8250 9.4947
Rule 29
7,791,745 7,791,745
0.7755 0.7781
Rule 30
578,539 584,644
4.6663 4.7025
Rule 31
565,385 569,840
6.6824 6.8606
Rule 32
4,725,662 4,725,662
1.0421 1.0472
Rule 33
611,109 611,575
9.1908 9.3105
Rule 34
5,927,236 5,927,236
0.8087 0.8116
Rule 35
574,524 575,271
5.3068 5.4356

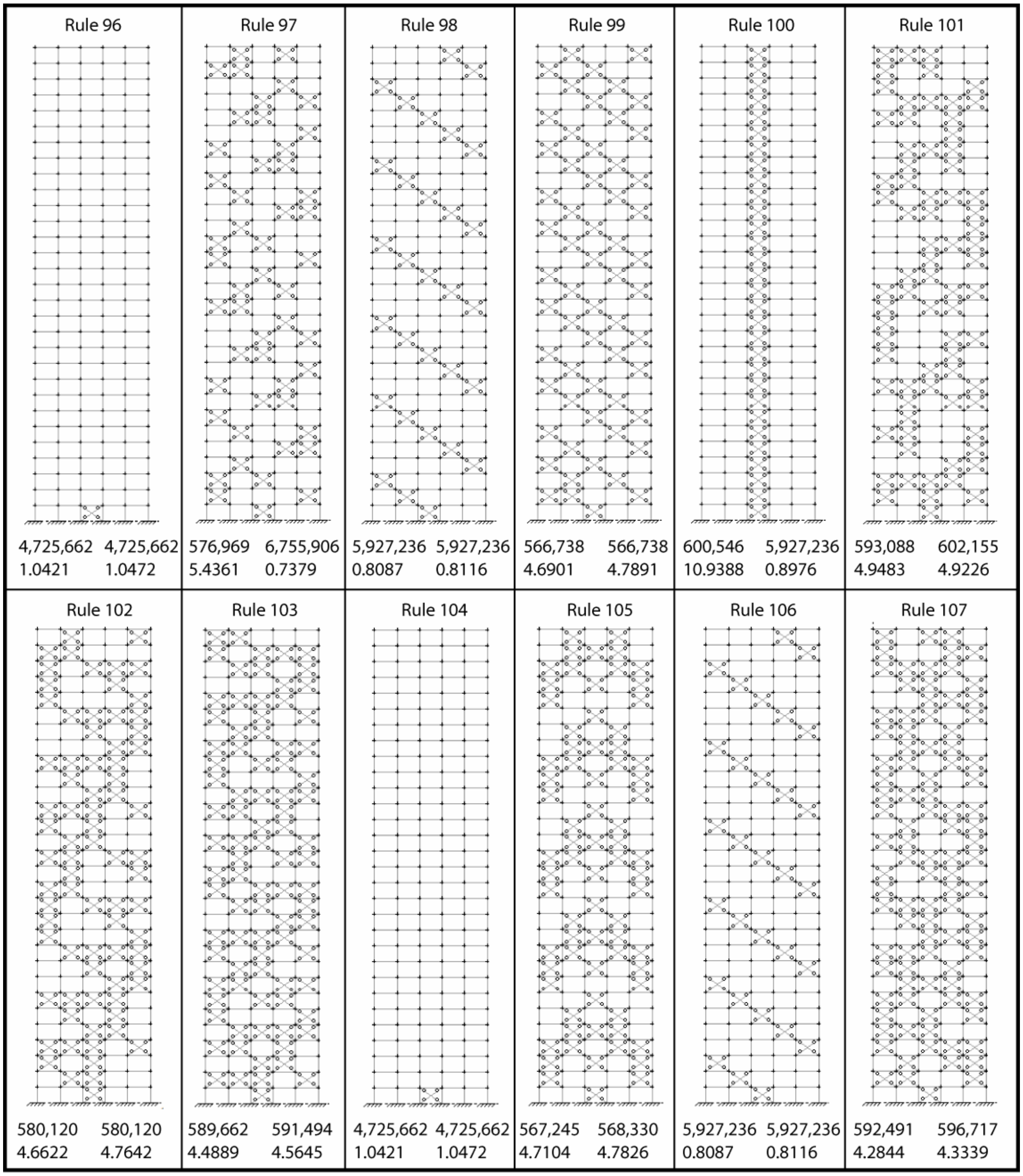

Rule 36
600,546 5,927,236
10.9388 0.8976
Rule 37
573,463 576,834
6.2483 6.4474
Rule 38
572,997 6,548,737
6.0351 0.7692
Rule 39
579,400 578,892
4.9619 5.0938
Rule 40
4,725,662 4,725,662
1.0421 1.0472
Rule 41
6,755,906 6,755,906
0.7383 0.7407
Rule 42
5,927,236 5,927,236
0.8087 0.8116
Rule 43
595,063 598,772
5.0474 5.1390
Rule 44
600,546 5,927,236
10.9388 0.8976
Rule 45
595,086 600,159
4.9604 4.9897
Rule 46
584,033 584,776
5.3575 5.4702
Rule 47
594,823 597,958
5.0063 5.1034

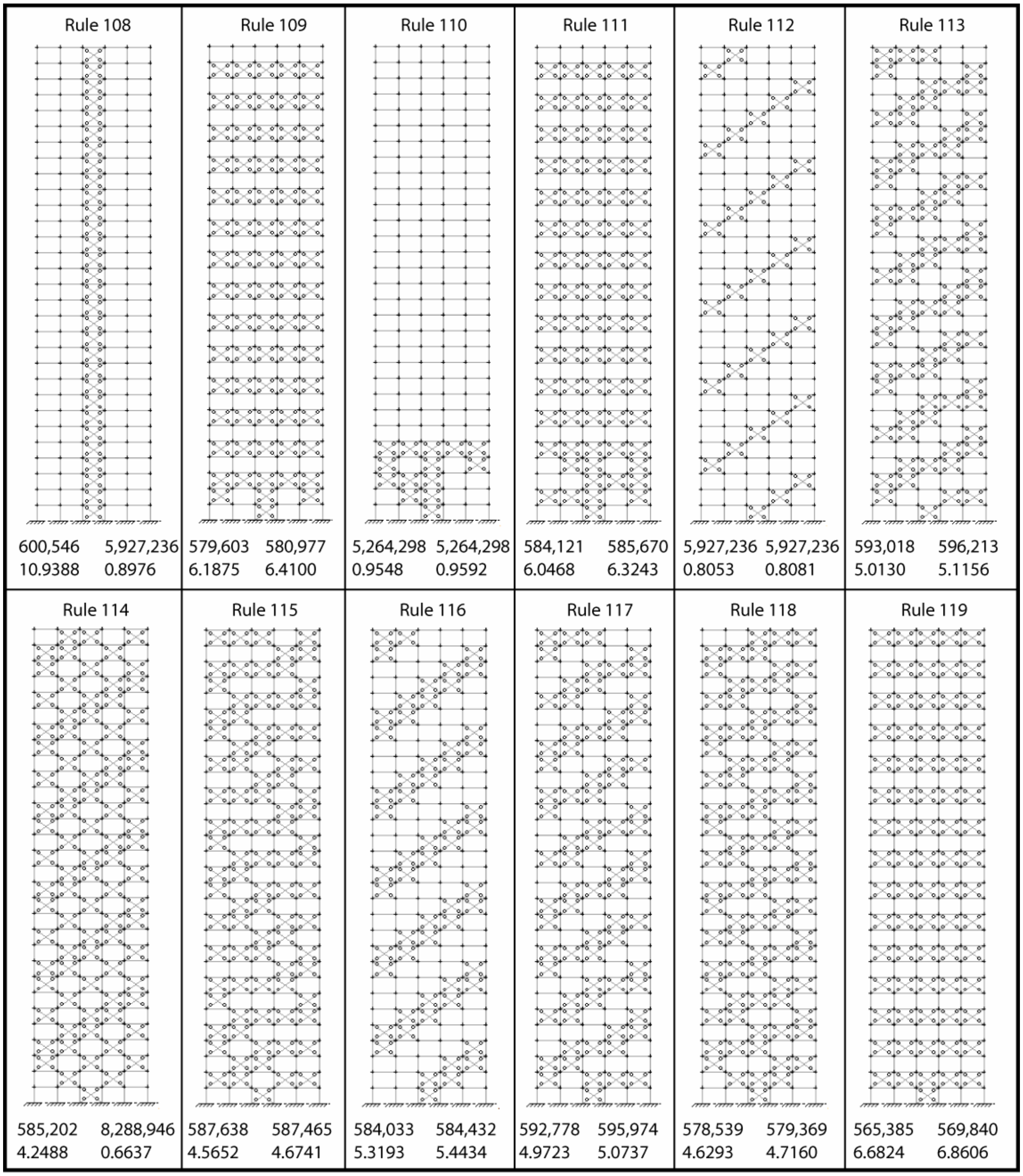

Rule 48
5,927,236 5,927,236
0.8053 0.8081
Rule 49
573,363 573,363
5.2786 5.4134
Rule 50
568,951 570,251
4.3405 4.4106
Rule 51
560,646 562,570
4.9940 5.1378
Rule 52
572,996 572,996
5.9890 6.1578
Rule 53
579,400 578,927
5.0393 5.1614
Rule 54
5,222,864 5,222,864
0.9415 0.9459
Rule 55
565,385 569,840
6.6824 6.8606
Rule 56
5,927,236 5,927,236
0.8053 0.8081
Rule 57
566,738 566,871
4.6311 4.7283
Rule 58
585,202 8,288,946
4.2916 0.6659
Rule 59
588,038 587,465
4.6196 4.7338


Rule 60
580,120 581,796
4.6955 4.7615
Rule 61
590,120 591,494
4.4956 4.5873
Rule 62
578,539 584,644
4.6663 4.7025
Rule 63
565,385 569,840
6.6824 6.8606
Rule 64
4,725,662 4,725,662
1.0421 1.0472
Rule 65
578,996 579,454
6.2270 6.5356
Rule 66
5,927,236 5,927,236
0.8087 0.8116
Rule 67
577,371 579,991
5.1441 5.2338
Rule 68
600,546 5,927,236
10.9388 0.8976
Rule 69
7,170,241 7,170,241
0.9329 0.9364
Rule 70
7,708,878 650,107
0.8213 9.5507
Rule 71
7,791,745 640,012
0.7717 7.8315

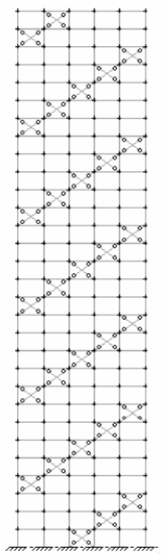

Rule 72
4,725,662 4,725,662
1.0421 1.0472
Rule 73
7,708,878 650,401
0.7871 9.0536
Rule 74
5,927,236 5,927,236
0.8087 0.8116
Rule 75
593,341 600,738
4.9490 4.9864
Rule 76
600,546 5,927,236
10.9388 0.8976
Rule 77
8,330,379 8,330,379
0.7645 0.7672
Rule 78
8,288,946 8,288,946
0.7737 0.7765
Rule 79
659,752 8,330,379
7.3025 0.7844
Rule 80
5,927,236 5,927,236
0.8053 0.8081
Rule 81
593,018 596,213
5.0130 5.1156
Rule 82
567,006 568671
5.1936 5.3053
Rule 83
584,817 589,509
5.0107 5.0842

Rule 84	Rule 85	Rule 86	Rule 87	Rule 88	Rule 89
					
584,033 584,432 5.3193 5.4434	581,884 588,157 5.1318 5.1782	578,539 579,369 4.6293 4.7160	565,385 569,840 6.6824 6.8606	5,927,236 5,927,236 0.8053 0.8081	7,791,743 600,738 0.7207 5.0339
Rule 90	**Rule 91**	**Rule 92**	**Rule 93**	**Rule 94**	**Rule 95**
					
7,874,611 7,874,611 0.7302 0.7326	579,623 580,714 6.0869 6.2710	8,288,946 8,288,946 0.7767 0.7794	8,330,379 8,330,379 0.7788 0.7814	8,330,379 8,330,379 0.7222 0.7246	565,385 569,840 6.6824 6.8606


Rule 96
4,725,662 4,725,662
1.0421 1.0472
Rule 97
576,969 6,755,906
5.4361 0.7379
Rule 98
5,927,236 5,927,236
0.8087 0.8116
Rule 99
566,738 566,738
4.6901 4.7891
Rule 100
600,546 5,927,236
10.9388 0.8976
Rule 101
593,088 602,155
4.9483 4.9226
Rule 102
580,120 580,120
4.6622 4.7642
Rule 103
589,662 591,494
4.4889 4.5645
Rule 104
4,725,662 4,725,662
1.0421 1.0472
Rule 105
567,245 568,330
4.7104 4.7826
Rule 106
5,927,236 5,927,236
0.8087 0.8116
Rule 107
592,491 596,717
4.2844 4.3339

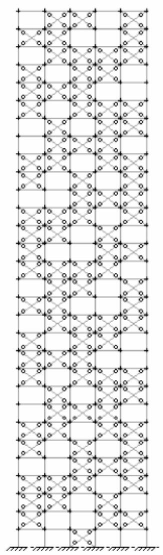

Rule 108
600,546 5,927,236
10.9388 0.8976
Rule 109
579,603 580,977
6.1875 6.4100
Rule 110
5,264,298 5,264,298
0.9548 0.9592
Rule 111
584,121 585,670
6.0468 6.3243
Rule 112
5,927,236 5,927,236
0.8053 0.8081
Rule 113
593,018 596,213
5.0130 5.1156
Rule 114
585,202 8,288,946
4.2488 0.6637
Rule 115
587,638 587,465
4.5652 4.6741
Rule 116
584,033 584,432
5.3193 5.4434
Rule 117
592,778 595,974
4.9723 5.0737
Rule 118
578,539 579,369
4.6293 4.7160
Rule 119
565,385 569,840
6.6824 6.8606

Rule 120



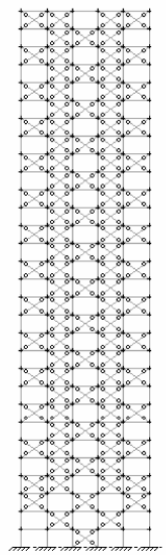
5,927,236 5,927,236
0.8053 0.8081

Rule 121



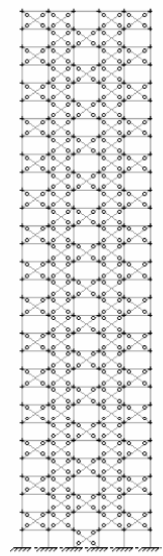
592,491 593,992
4.2632 4.3304

Rule 122



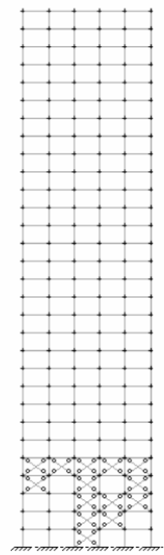
601,682 600,651
4.3351 4.4224

Rule 123



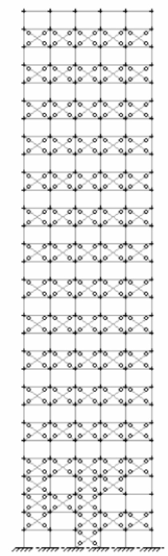
605,588 607,811
4.3186 4.3713

Rule 124



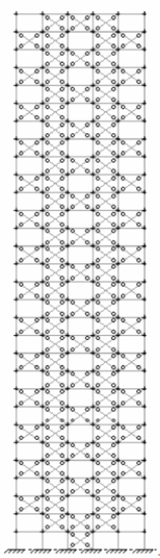
5,264,298 5,264,298
0.9595 0.9639

Rule 125



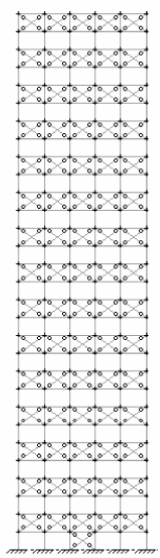
584,121 585,670
6.0200 6.2975

Rule 126



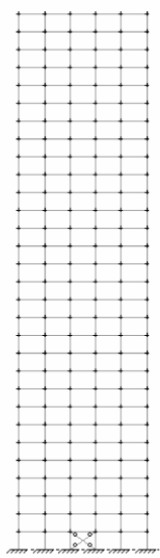
602,305 602,305
4.3327 4.4193

Rule 127



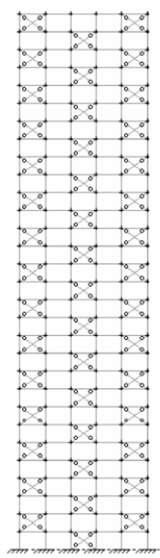
565,385 569,840
6.6824 6.8606

Rule 128



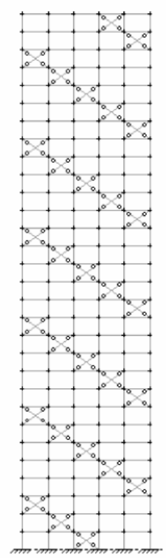
4,725,662 4,725,662
1.0421 1.0472

Rule 129



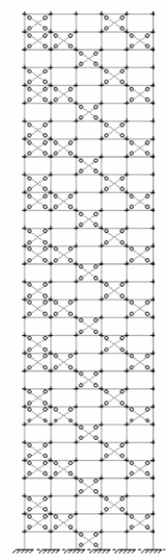
611,109 611,575
9.1908 9.3105

Rule 130

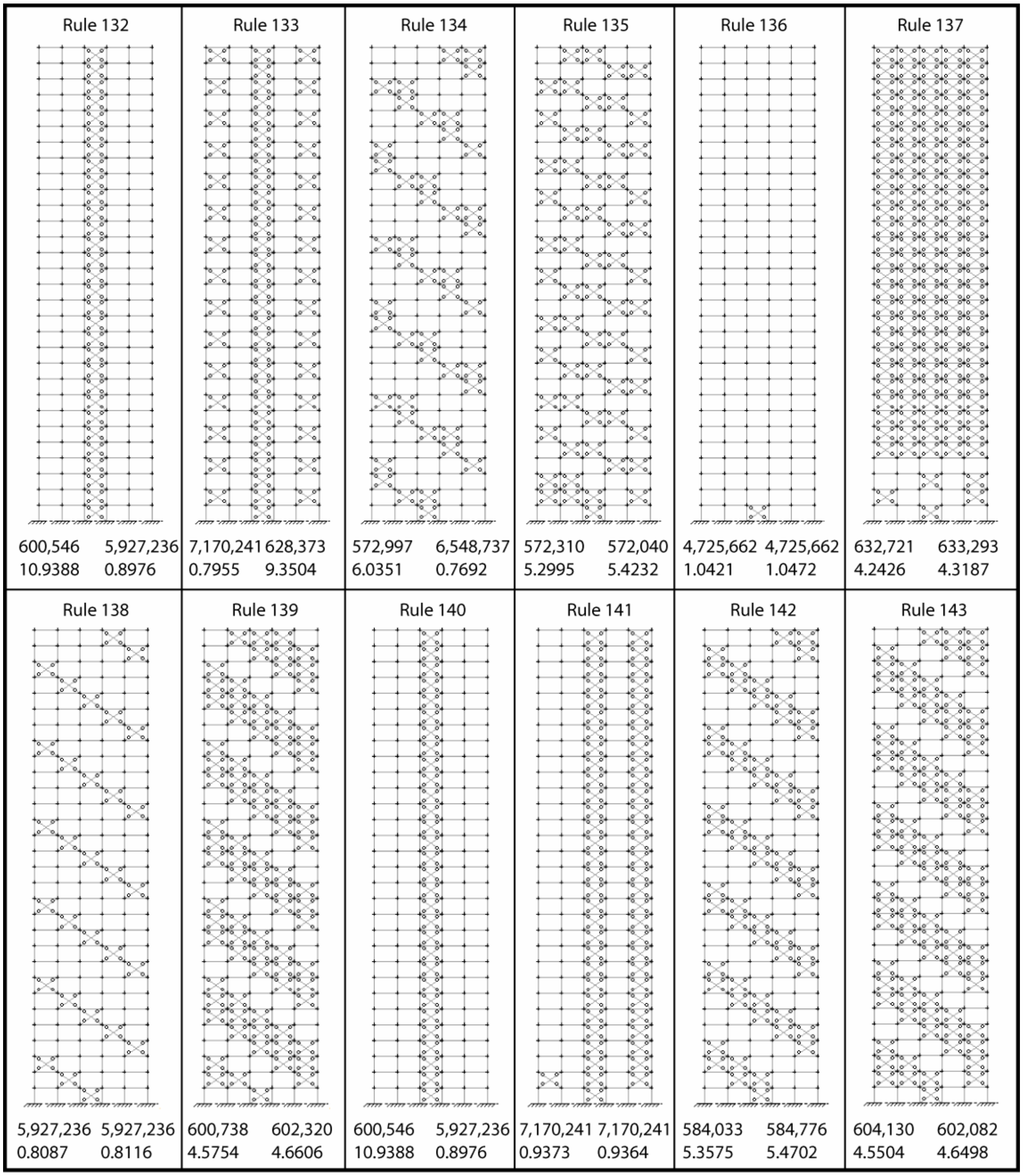


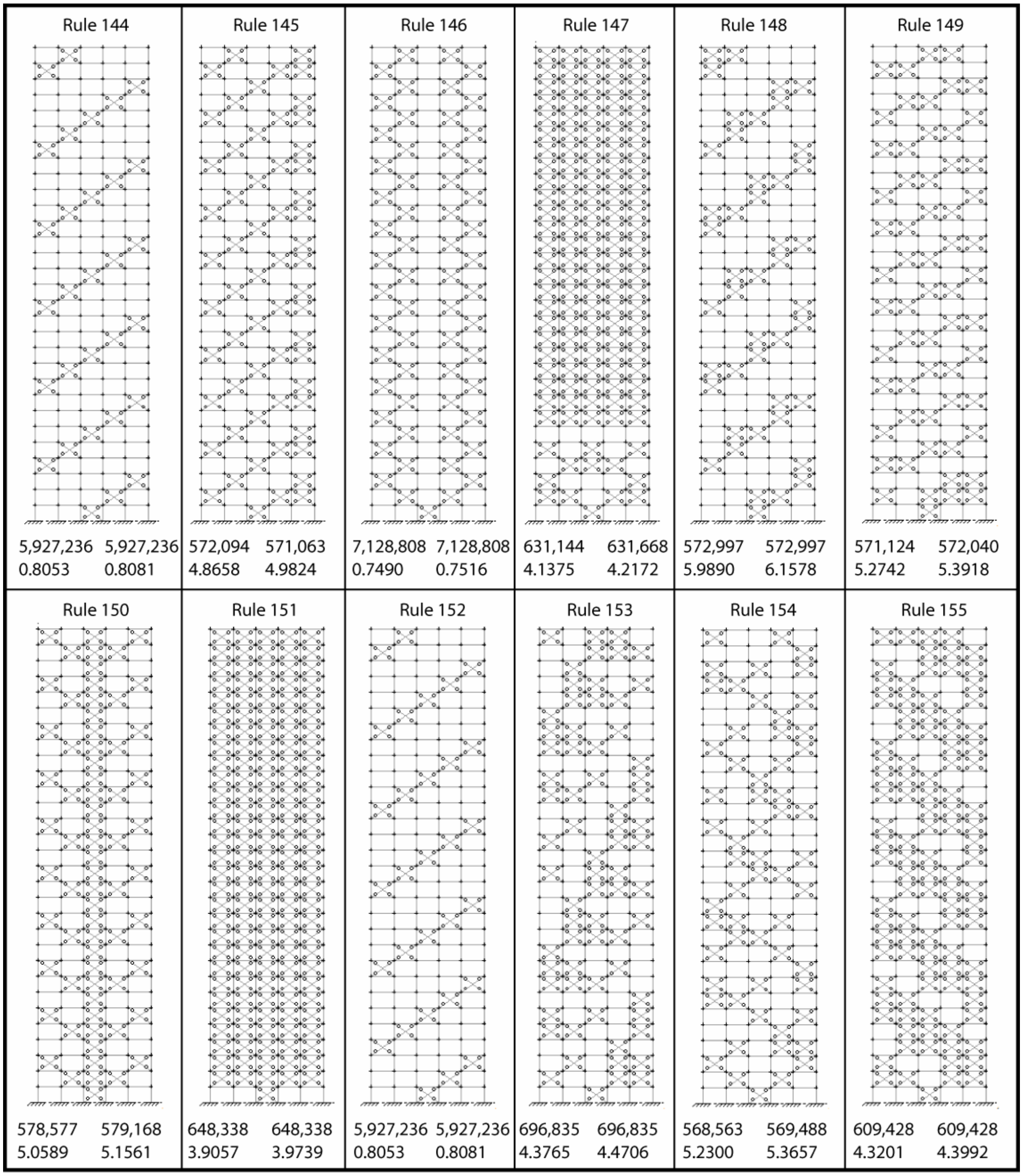
5,927,236 5,927,236
0.8087 0.8116

Rule 131

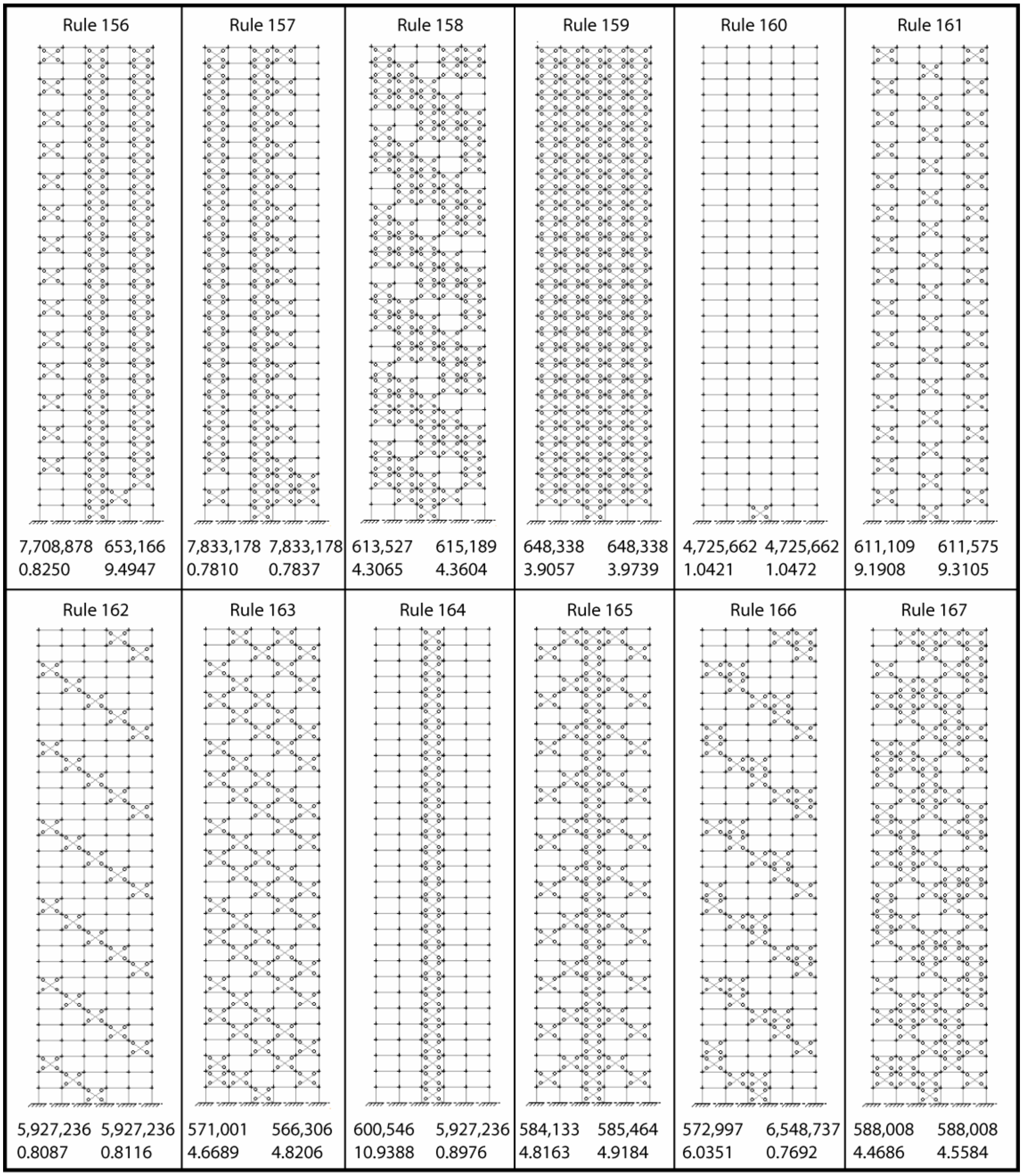


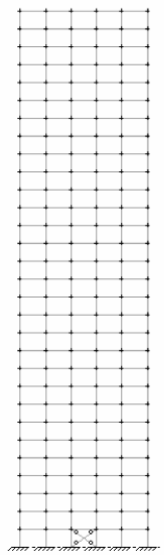
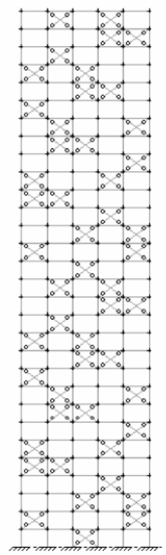
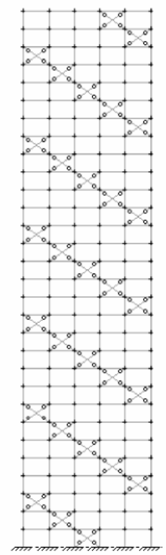
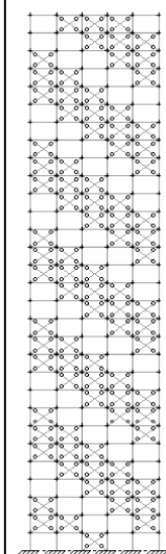
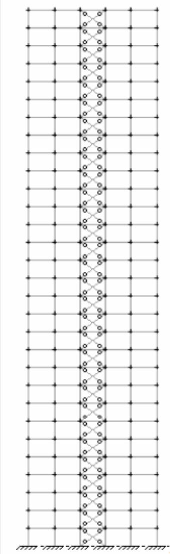
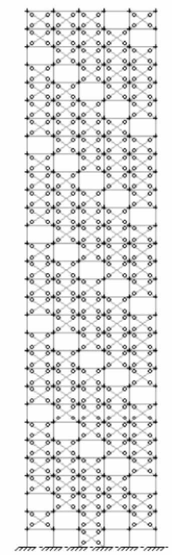
573,185 572,154
4.9178 5.0352

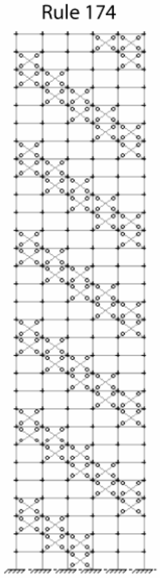
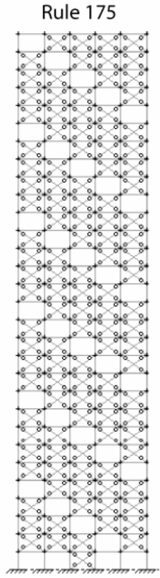
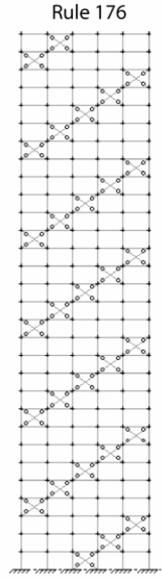
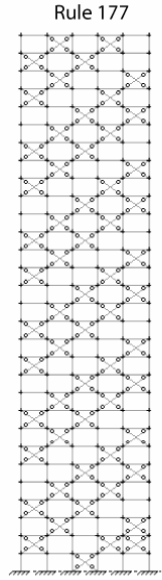
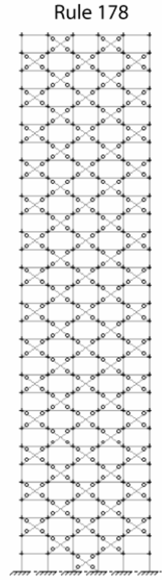
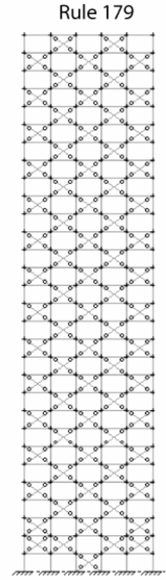

Rule 132
600,546 5,927,236
10.9388 0.8976
Rule 133
7,170,241 628,373
0.7955 9.3504
Rule 134
572,997 6,548,737
6.0351 0.7692
Rule 135
572,310 572,040
5.2995 5.4232
Rule 136
4,725,662 4,725,662
1.0421 1.0472
Rule 137
632,721 633,293
4.2426 4.3187
Rule 138
5,927,236 5,927,236
0.8087 0.8116
Rule 139
600,738 602,320
4.5754 4.6606
Rule 140
600,546 5,927,236
10.9388 0.8976
Rule 141
7,170,241 7,170,241
0.9373 0.9364
Rule 142
584,033 584,776
5.3575 5.4702
Rule 143
604,130 602,082
4.5504 4.6498

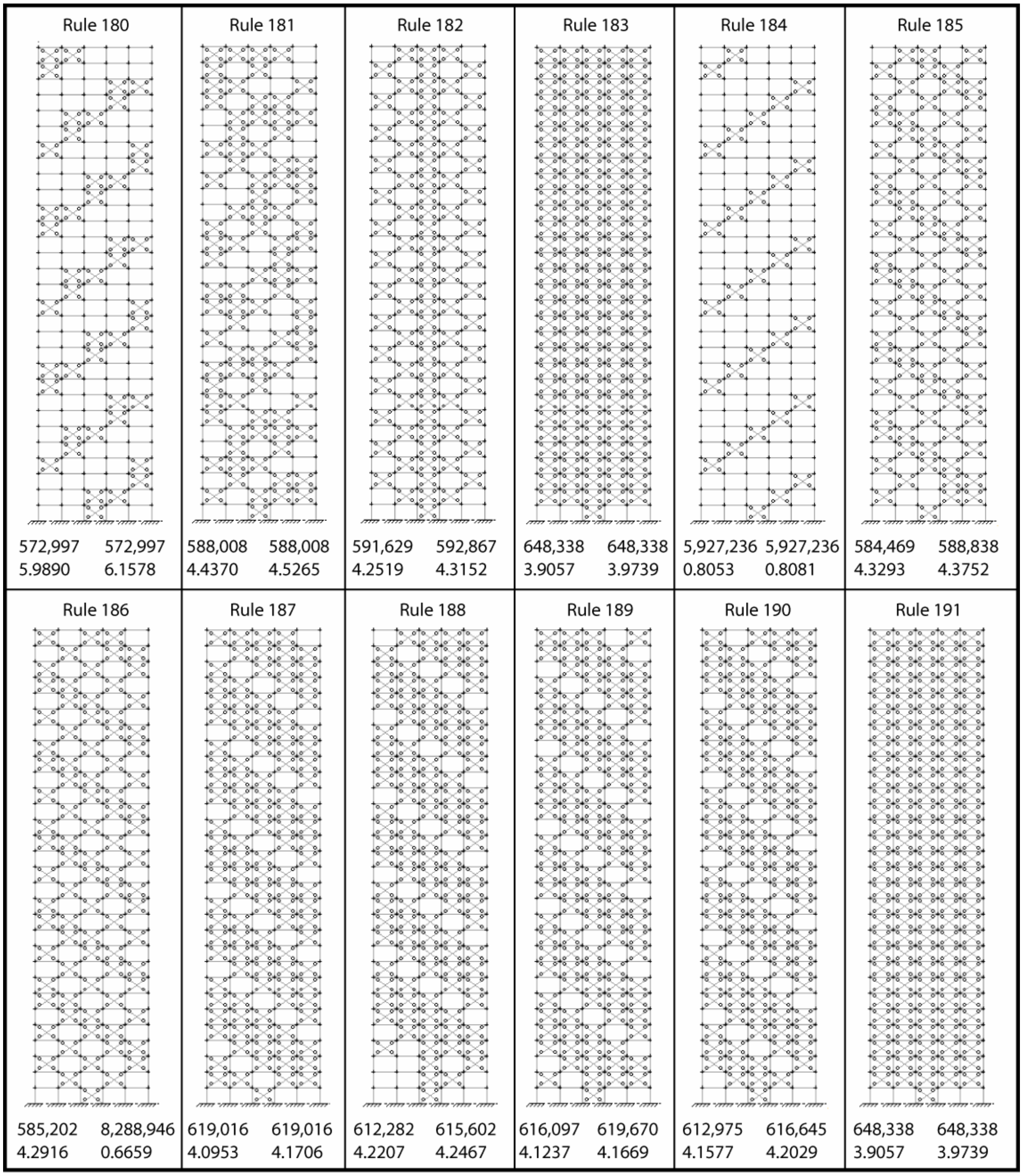


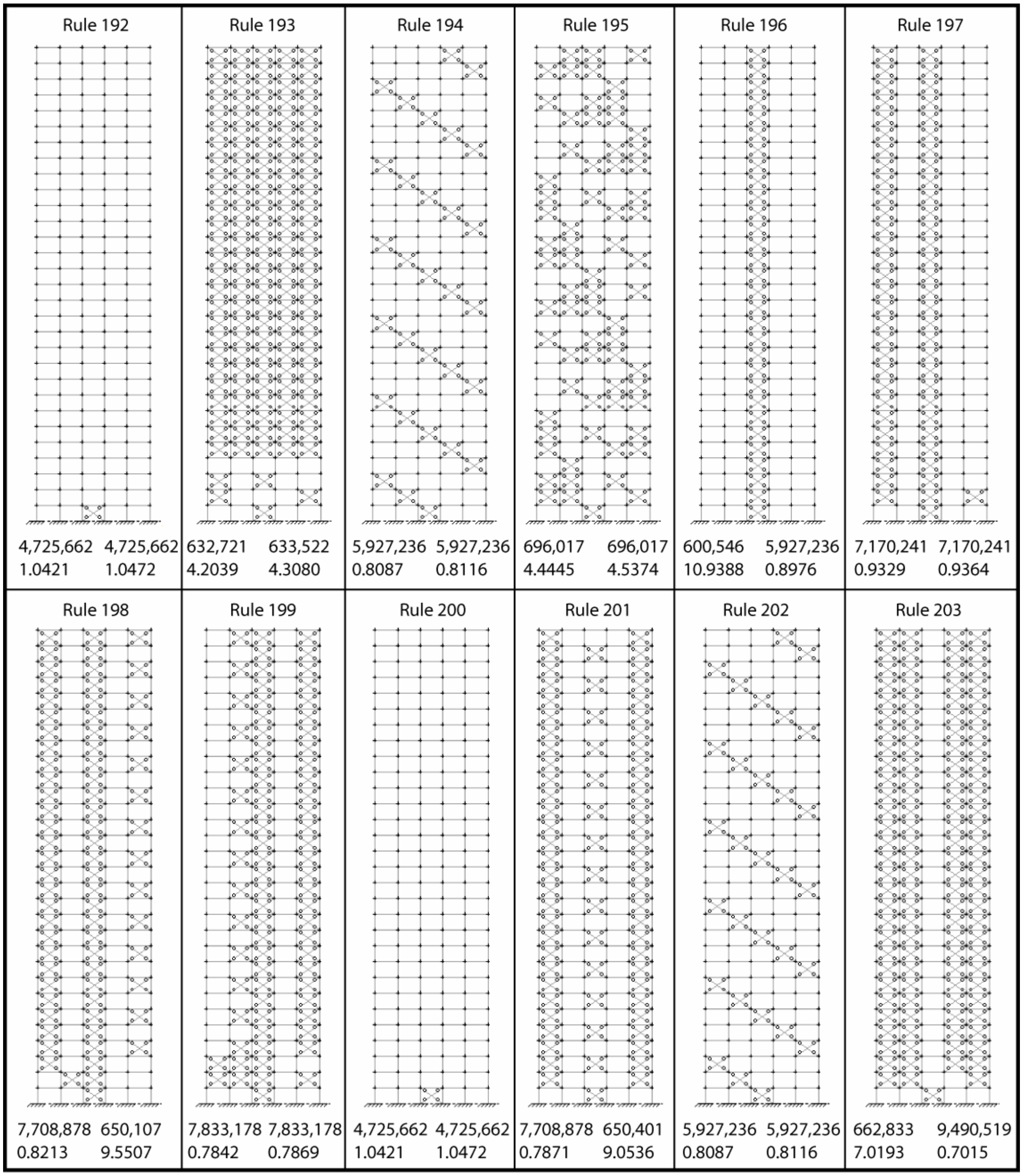
Rule 144
5,927,236 5,927,236
0.8053 0.8081
Rule 145
572,094 571,063
4.8658 4.9824
Rule 146
7,128,808 7,128,808
0.7490 0.7516
Rule 147
631,144 631,668
4.1375 4.2172
Rule 148
572,997 572,997
5.9890 6.1578
Rule 149
571,124 572,040
5.2742 5.3918
Rule 150
578,577 579,168
5.0589 5.1561
Rule 151
648,338 648,338
3.9057 3.9739
Rule 152
5,927,236 5,927,236
0.8053 0.8081
Rule 153
696,835 696,835
4.3765 4.4706
Rule 154
568,563 569,488
5.2300 5.3657
Rule 155
609,428 609,428
4.3201 4.3992

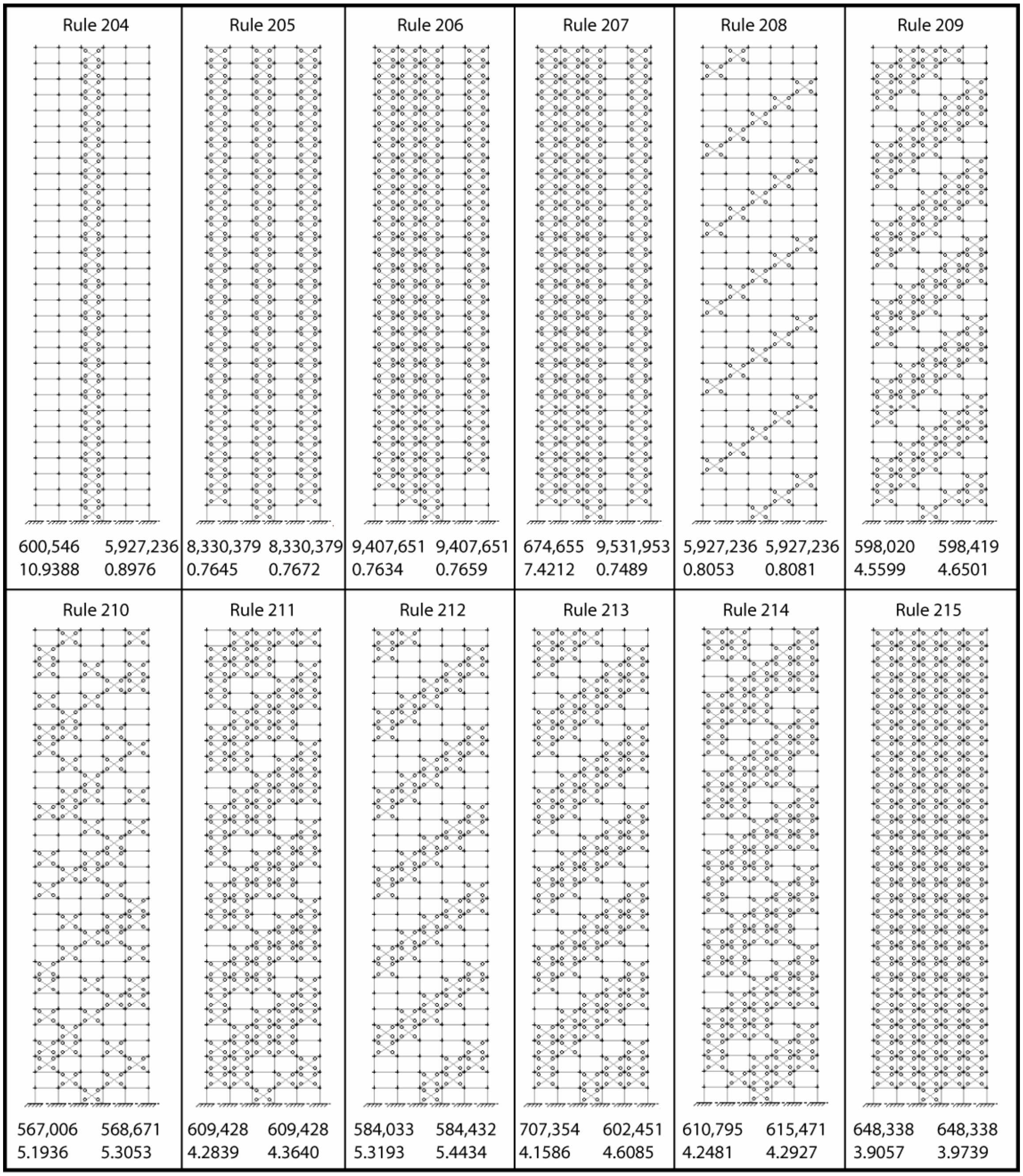

Rule 156
7,708,878 653,166
0.8250 9.4947
Rule 157
7,833,178 7,833,178
0.7810 0.7837
Rule 158
613,527 615,189
4.3065 4.3604
Rule 159
648,338 648,338
3.9057 3.9739
Rule 160
4,725,662 4,725,662
1.0421 1.0472
Rule 161
611,109 611,575
9.1908 9.3105
Rule 162
5,927,236 5,927,236
0.8087 0.8116
Rule 163
571,001 566,306
4.6689 4.8206
Rule 164
600,546 5,927,236
10.9388 0.8976
Rule 165
584,133 585,464
4.8163 4.9184
Rule 166
572,997 6,548,737
6.0351 0.7692
Rule 167
588,008 588,008
4.4686 4.5584

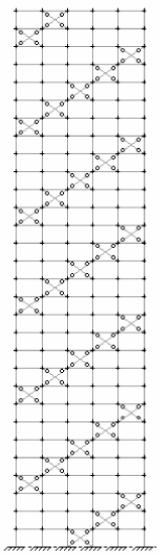
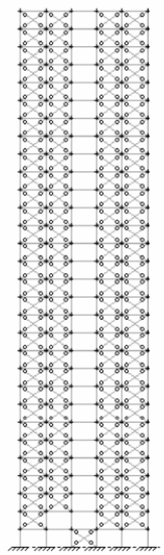
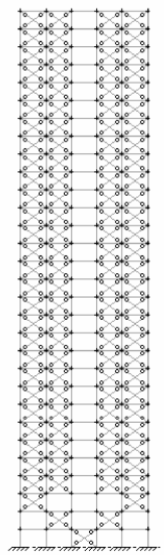
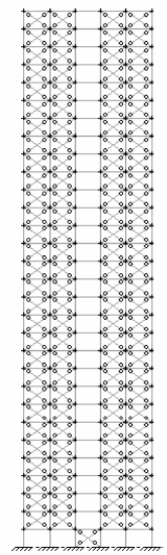
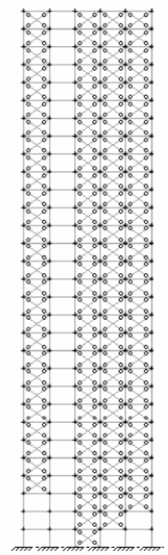
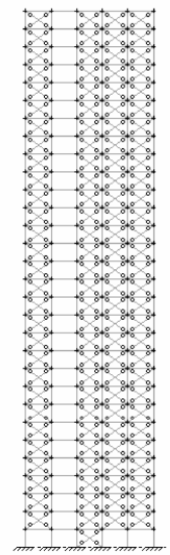
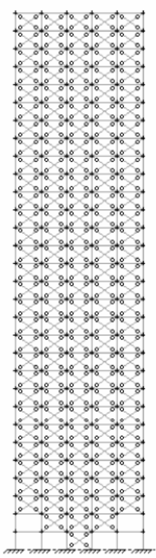
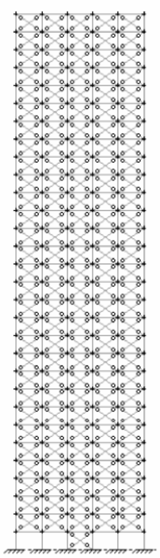
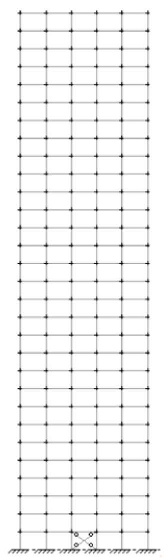
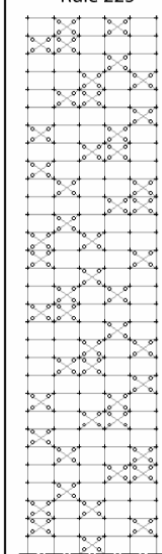
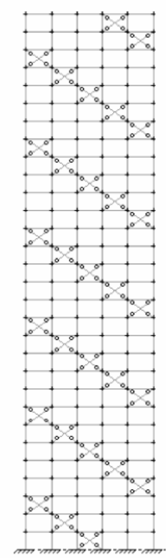
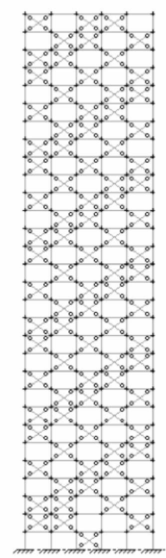
Rule 168	Rule 169	Rule 170	Rule 171	Rule 172	Rule 173
					
4,725,662 4,725,662 1.0421 1.0472	6,755,906 6,755,906 0.7383 0.7407	5,927,236 5,927,236 0.8087 0.8116	600,738 602,320 4.5754 4.6606	600,546 5,927,236 10.9388 0.8976	614,159 614,159 4.1242 4.1998

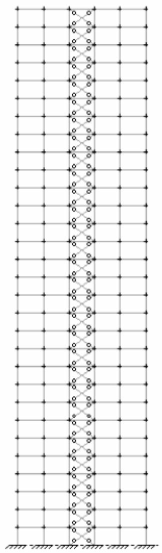
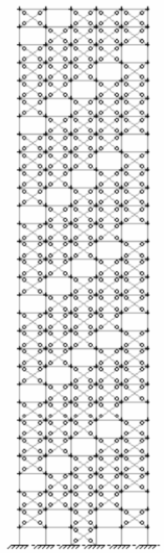
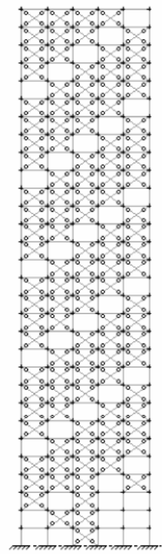
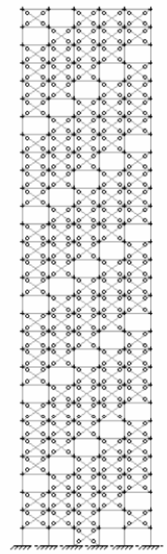
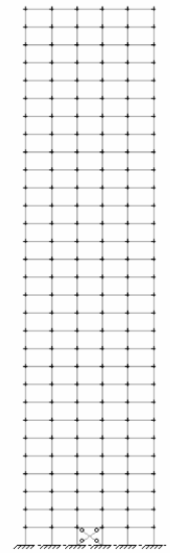
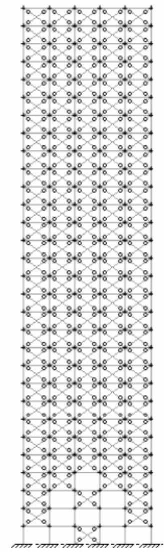
Rule 174	Rule 175	Rule 176	Rule 177	Rule 178	Rule 179
					
584,033 584,776 5.3575 5.4702	616,673 616,673 4.1147 4.1904	5,927,236 5,927,236 0.8053 0.8081	569,840 565,145 4.6401 4.7913	568,951 570,251 4.3405 4.4106	571,059 571,059 4.3232 4.4088

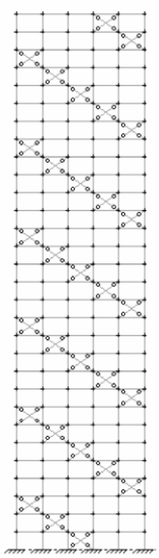
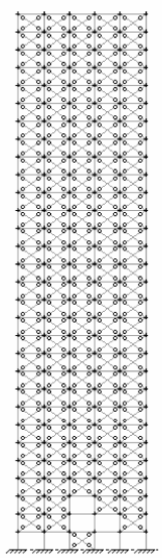
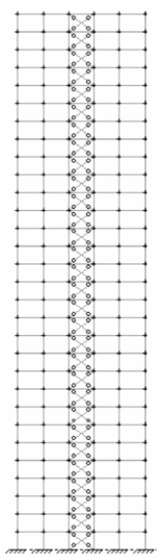
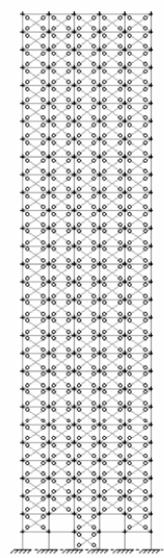
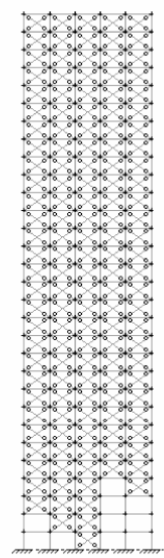
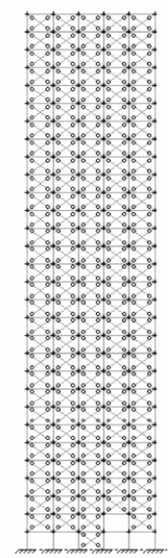

Rule 180
572,997 572,997
5.9890 6.1578
Rule 181
588,008 588,008
4.4370 4.5265
Rule 182
591,629 592,867
4.2519 4.3152
Rule 183
648,338 648,338
3.9057 3.9739
Rule 184
5,927,236 5,927,236
0.8053 0.8081
Rule 185
584,469 588,838
4.3293 4.3752
Rule 186
585,202 8,288,946
4.2916 0.6659
Rule 187
619,016 619,016
4.0953 4.1706
Rule 188
612,282 615,602
4.2207 4.2467
Rule 189
616,097 619,670
4.1237 4.1669
Rule 190
612,975 616,645
4.1577 4.2029
Rule 191
648,338 648,338
3.9057 3.9739

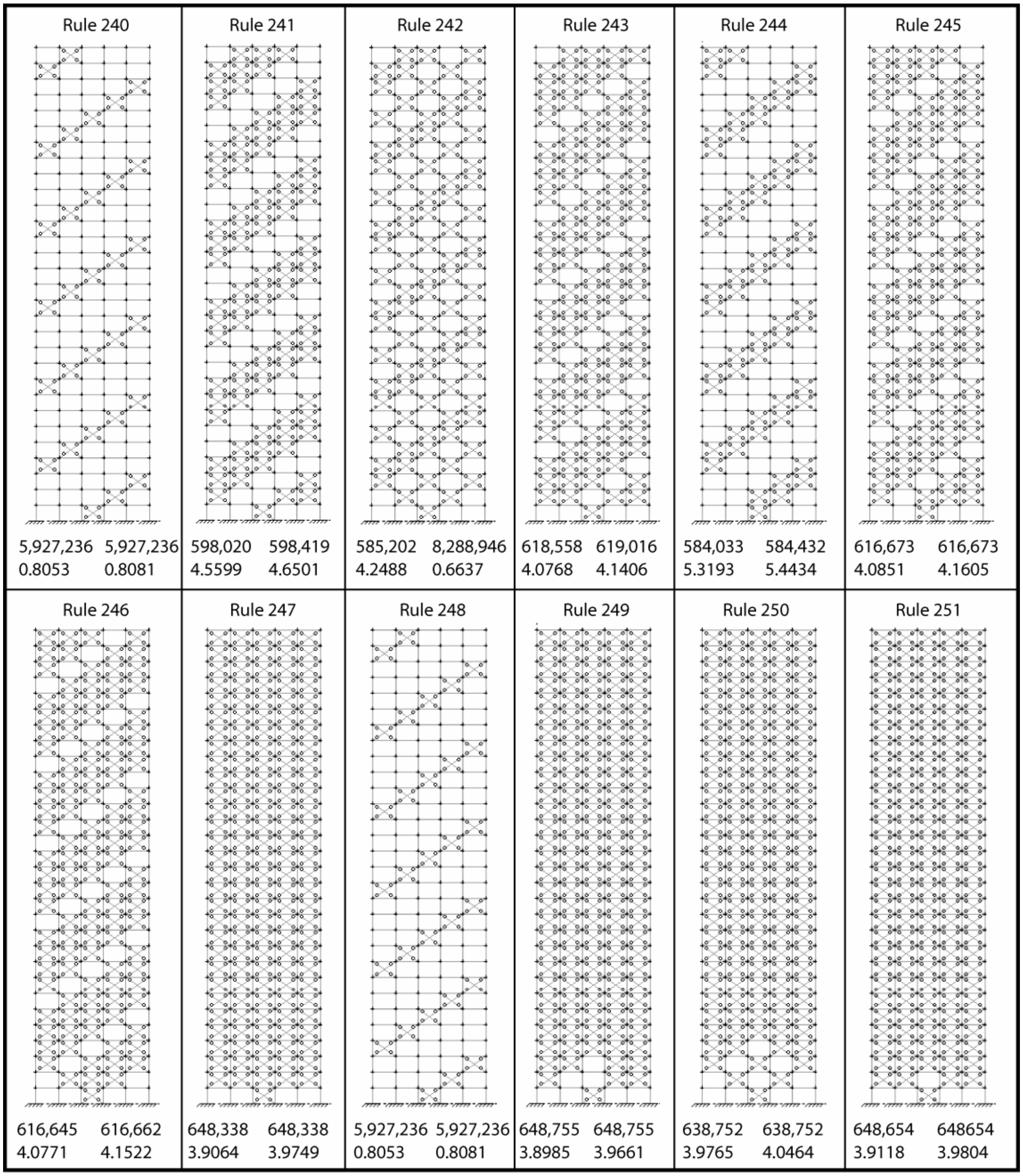

Rule 192
4,725,662 4,725,662
1.0421 1.0472
Rule 193
632,721 633,522
4.2039 4.3080
Rule 194
5,927,236 5,927,236
0.8087 0.8116
Rule 195
696,017 696,017
4.4445 4.5374
Rule 196
600,546 5,927,236
10.9388 0.8976
Rule 197
7,170,241 7,170,241
0.9329 0.9364
Rule 198
7,708,878 650,107
0.8213 9.5507
Rule 199
7,833,178 7,833,178
0.7842 0.7869
Rule 200
4,725,662 4,725,662
1.0421 1.0472
Rule 201
7,708,878 650,401
0.7871 9.0536
Rule 202
5,927,236 5,927,236
0.8087 0.8116
Rule 203
662,833 9,490,519
7.0193 0.7015

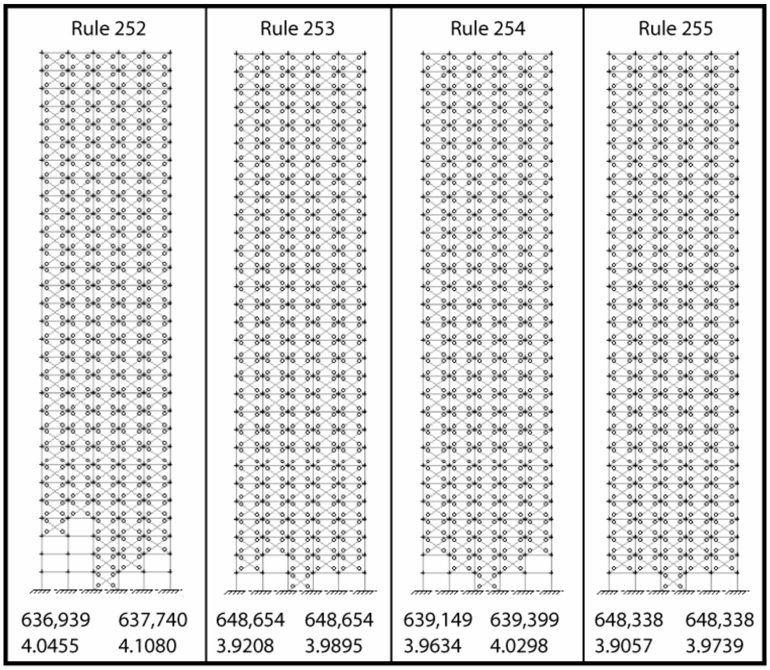

Rule 204
600,546 5,927,236
10.9388 0.8976
Rule 205
8,330,379 8,330,379
0.7645 0.7672
Rule 206
9,407,651 9,407,651
0.7634 0.7659
Rule 207
674,655 9,531,953
7.4212 0.7489
Rule 208
5,927,236 5,927,236
0.8053 0.8081
Rule 209
598,020 598,419
4.5599 4.6501
Rule 210
567,006 568,671
5.1936 5.3053
Rule 211
609,428 609,428
4.2839 4.3640
Rule 212
584,033 584,432
5.3193 5.4434
Rule 213
707,354 602,451
4.1586 4.6085
Rule 214
610,795 615,471
4.2481 4.2927
Rule 215
648,338 648,338
3.9057 3.9739

Rule 216	Rule 217	Rule 218	Rule 219	Rule 220	Rule 221
					
5,927,236 5,927,236 0.8053 0.8081	664,742 9,490,519 6.8783 0.7038	9,366,218 9,366,218 0.7084 0.7107	9,531,953 9,531,953 0.6987 0.7010	9,407,651 9,407,651 0.7664 0.7688	9,531,953 9,531,953 0.7491 0.7515
Rule 222	**Rule 223**	**Rule 224**	**Rule 225**	**Rule 226**	**Rule 227**
					
639,149 639,399 3.9634 4.0298	648,338 648,338 3.9057 3.9739	4,725,662 4,725,662 1.0421 1.0472	576,969 6,755,906 5.4361 0.7379	5,927,236 5,927,236 0.8087 0.8116	584,469 584,469 4.2896 4.3730

Rule 228	Rule 229	Rule 230	Rule 231	Rule 232	Rule 233
					
600,546 5,927,236 10.9388 0.8976	614,159 614,159 4.0724 4.1476	612,282 615,602 4.1911 4.2169	616,555 616,555 4.0616 4.1366	4,725,662 4,725,662 1.0421 1.0472	641,172 641,172 3.9734 4.0430

Rule 234	Rule 235	Rule 236	Rule 237	Rule 238	Rule 239
					
5,927,236 5,927,236 0.8087 0.8116	648,555 648,755 3.8923 3.9583	600,546 5,927,236 10.9388 0.8976	646,308 647,009 3.9305 3.9949	636,939 636,939 4.0069 4.0782	648,655 648,655 3.9035 3.9717


Rule 240
5,927,236 5,927,236
0.8053 0.8081
Rule 241
598,020 598,419
4.5599 4.6501
Rule 242
585,202 8,288,946
4.2488 0.6637
Rule 243
618,558 619,016
4.0768 4.1406
Rule 244
584,033 584,432
5.3193 5.4434
Rule 245
616,673 616,673
4.0851 4.1605
Rule 246
616,645 616,662
4.0771 4.1522
Rule 247
648,338 648,338
3.9064 3.9749
Rule 248
5,927,236 5,927,236
0.8053 0.8081
Rule 249
648,755 648,755
3.8985 3.9661
Rule 250
638,752 638,752
3.9765 4.0464
Rule 251
648,654 648654
3.9118 3.9804

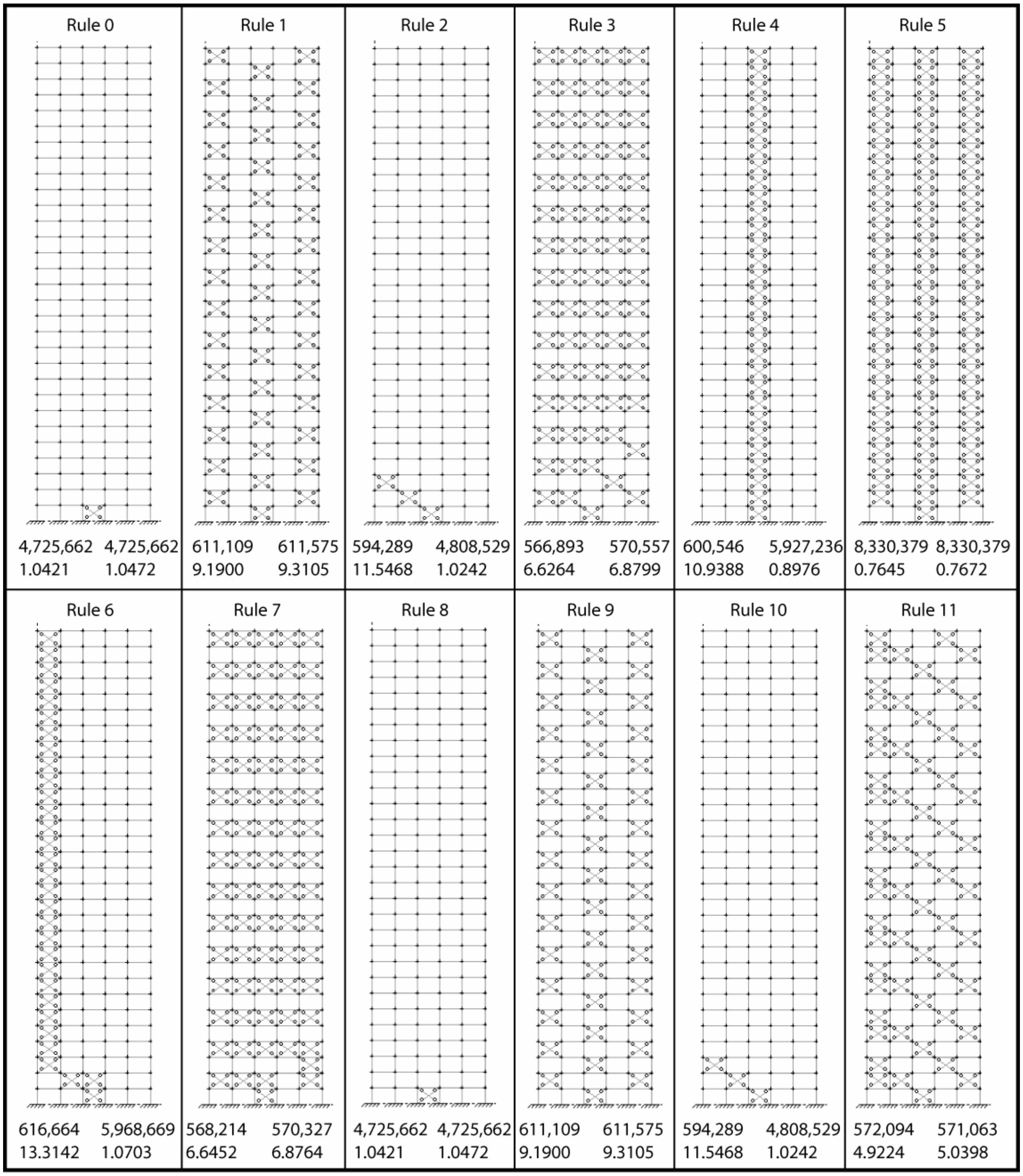

Rule 252
636,939 637,740
4.0455 4.1080
Rule 253
648,654 648,654
3.9208 3.9895
Rule 254
639,149 639,399
3.9634 4.0298
Rule 255
648,338 648,338
3.9057 3.9739

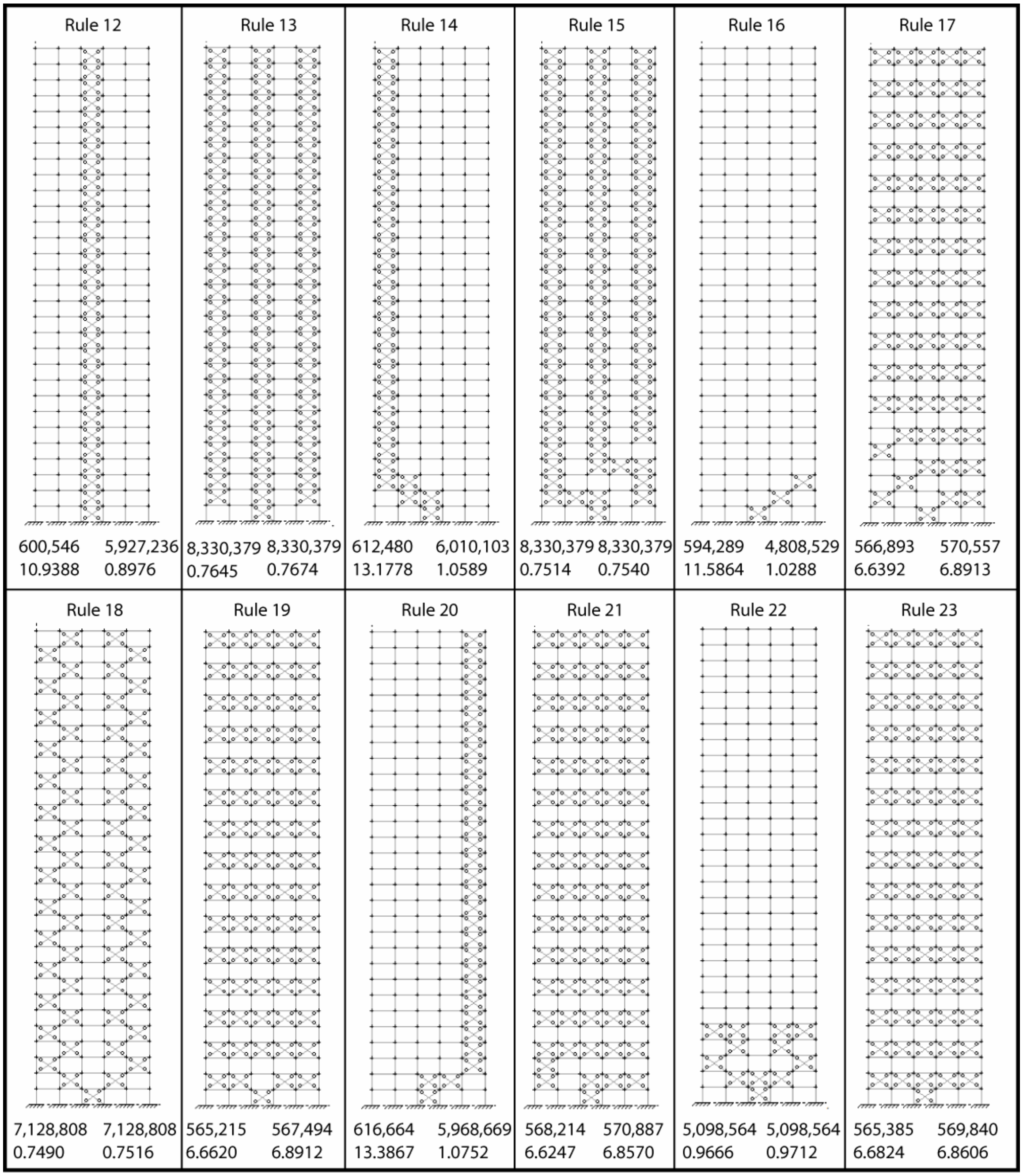
APPENDIX C

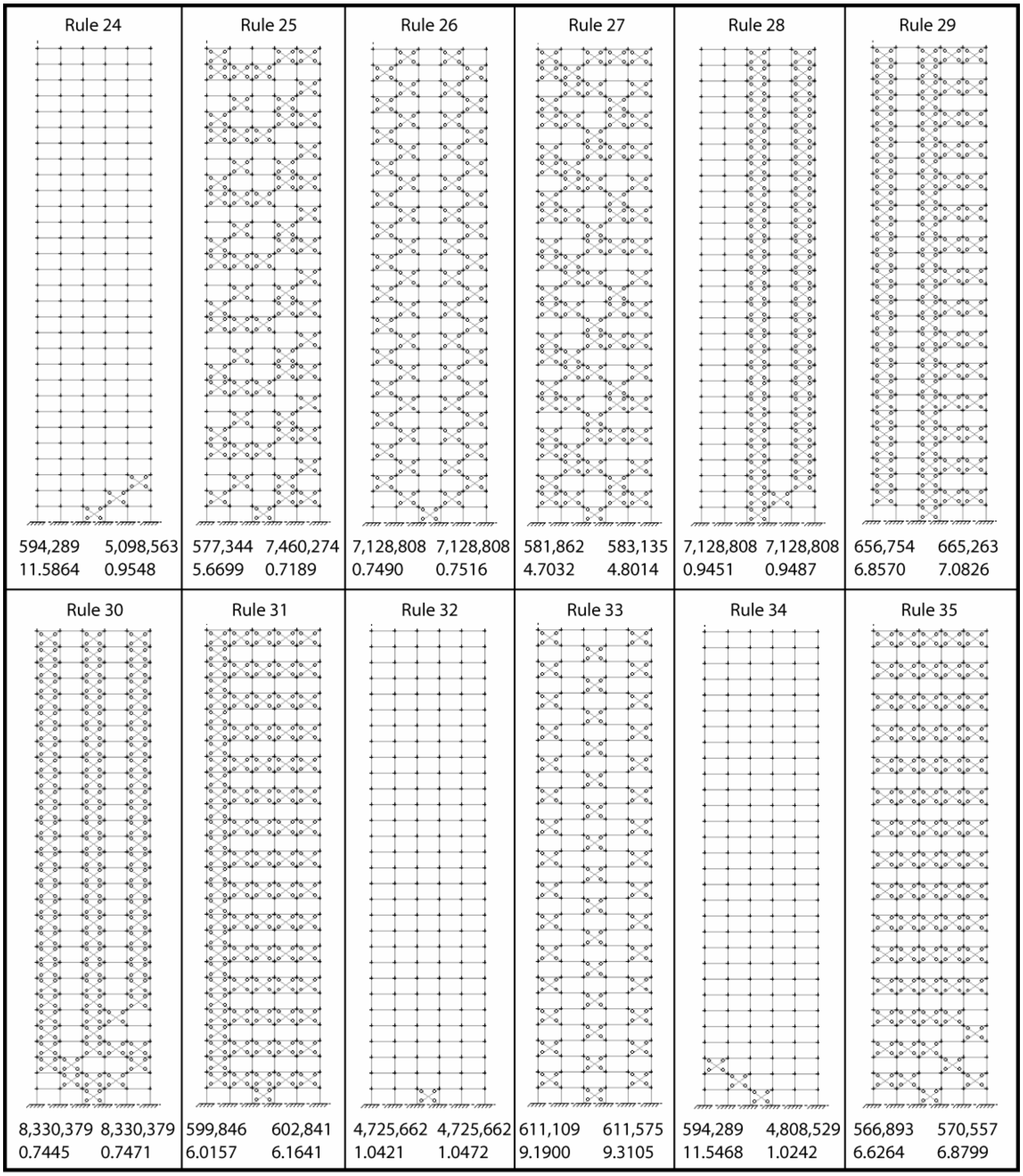
In this appendix, the entire set of 256 design concepts of wind bracing systems generated by elementary CAs with nonperiodic boundary conditions is presented. As in Appendix A, the designs have been generated from the simplest design embryo consisting of a single X bracing located in the central bay.

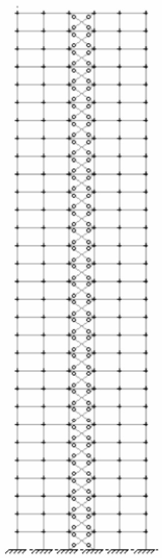
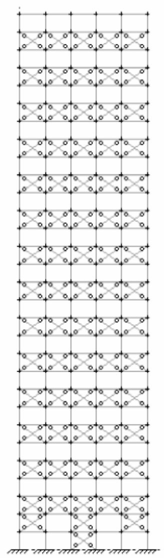
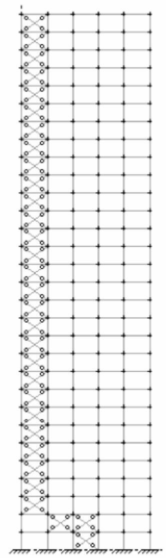
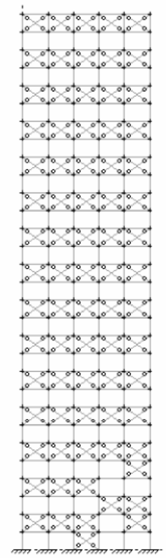
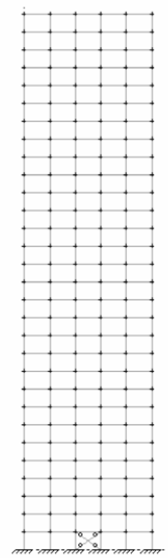
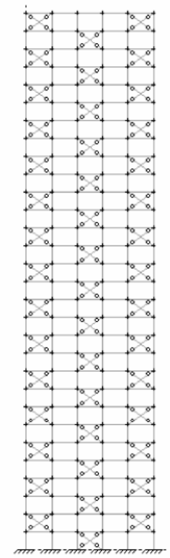
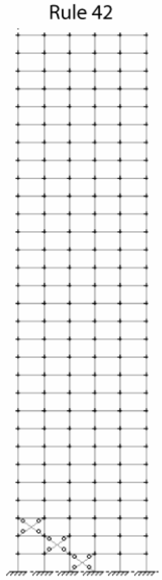
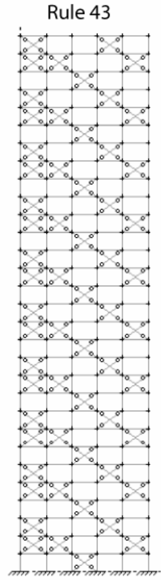
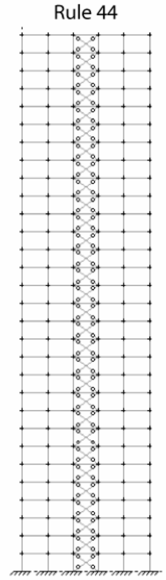
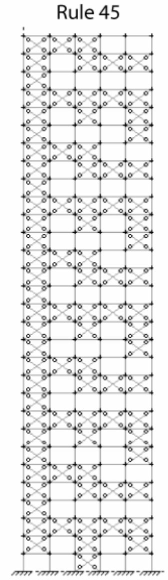
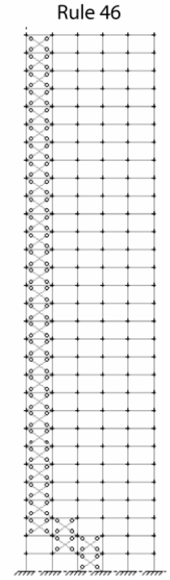
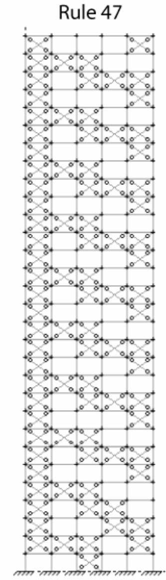
Each cell in this table contains the number of a design rule at the top, the actual design developed from the design embryo by this rule (center), and four values arranged in a 2 x 2 array (the bottom part) as shown on the right. This array contains four values representing the total weight of the steel structural system (the first row) and its maximum horizontal displacement (the second row). The first column contains measurements obtained using the first-order structural analysis while the second column contains the values produced by a more accurate and at the same time more computationally expensive P-Δ analysis. The values of the total weight of the steel structural system presented in the first row are measured in lbs. whereas the values of the maximum horizontal displacement, shown in the second row, are measured in inches.

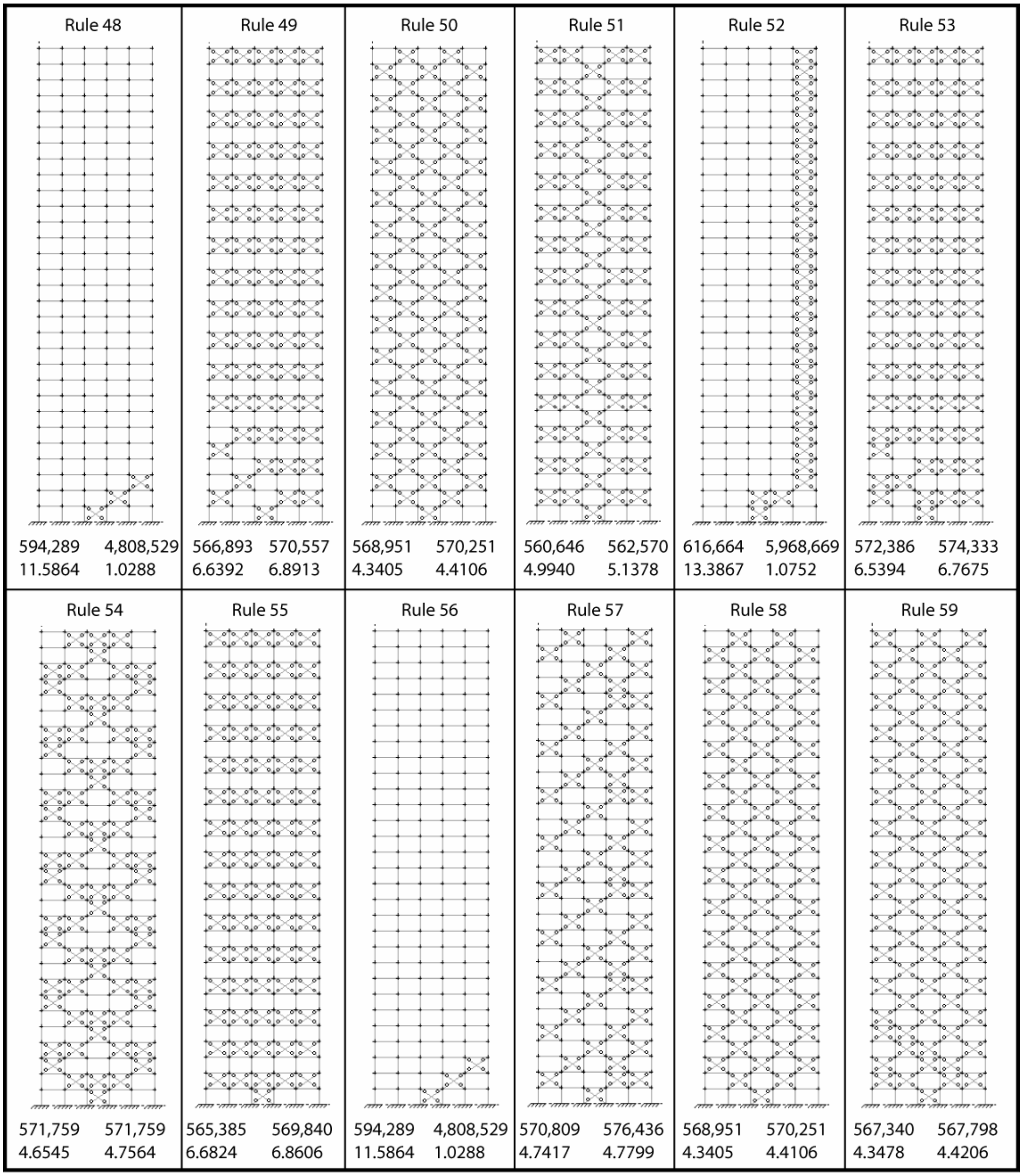
Total weight of the steel structure (1st-order)	Total weight of the steel structure (P-Delta)
Maximum horizontal displacement (1st-order)	Maximum horizontal displacement (P-Delta)


Rule 0
4,725,662 4,725,662
1.0421 1.0472
Rule 1
611,109 611,575
9.1900 9.3105
Rule 2
594,289 4,808,529
11.5468 1.0242
Rule 3
566,893 570,557
6.6264 6.8799
Rule 4
600,546 5,927,236
10.9388 0.8976
Rule 5
8,330,379 8,330,379
0.7645 0.7672
Rule 6
616,664 5,968,669
13.3142 1.0703
Rule 7
568,214 570,327
6.6452 6.8764
Rule 8
4,725,662 4,725,662
1.0421 1.0472
Rule 9
611,109 611,575
9.1900 9.3105
Rule 10
594,289 4,808,529
11.5468 1.0242
Rule 11
572,094 571,063
4.9224 5.0398

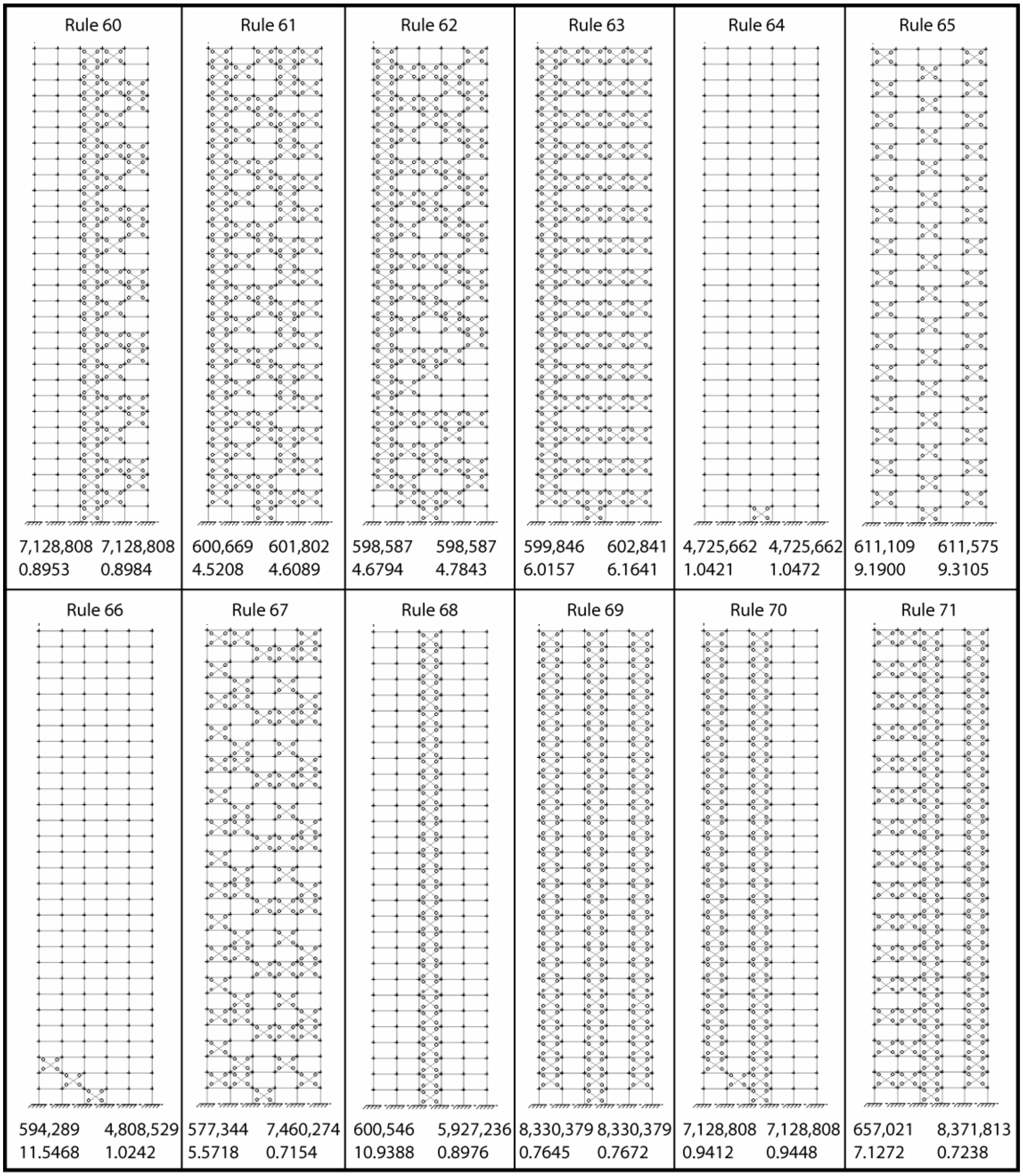

Rule 12
600,546 5,927,236
10.9388 0.8976
Rule 13
8,330,379 8,330,379
0.7645 0.7674
Rule 14
612,480 6,010,103
13.1778 1.0589
Rule 15
8,330,379 8,330,379
0.7514 0.7540
Rule 16
594,289 4,808,529
11.5864 1.0288
Rule 17
566,893 570,557
6.6392 6.8913
Rule 18
7,128,808 7,128,808
0.7490 0.7516
Rule 19
565,215 567,494
6.6620 6.8912
Rule 20
616,664 5,968,669
13.3867 1.0752
Rule 21
568,214 570,887
6.6247 6.8570
Rule 22
5,098,564 5,098,564
0.9666 0.9712
Rule 23
565,385 569,840
6.6824 6.8606

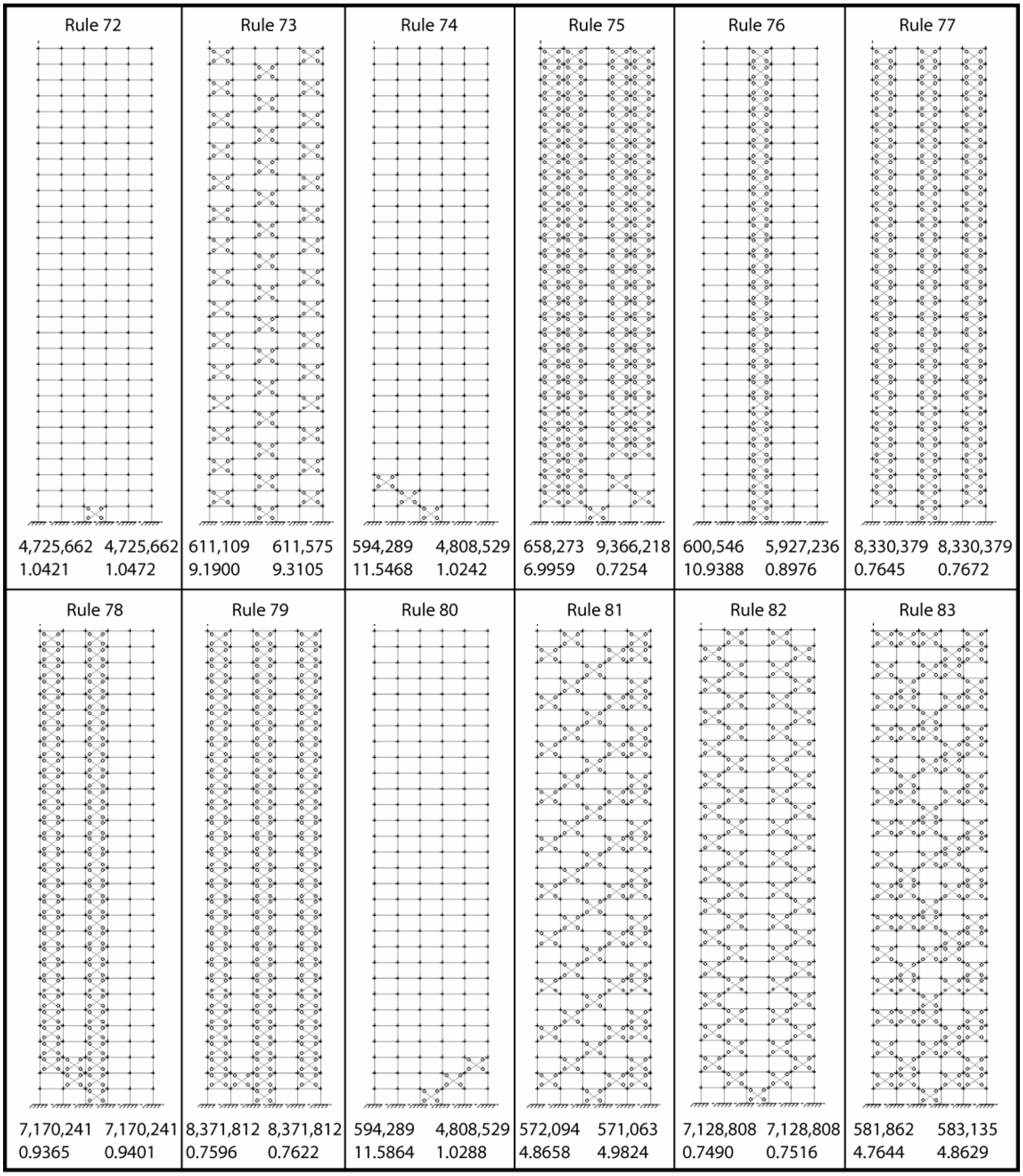

Rule 24
594,289 5,098,563
11.5864 0.9548
Rule 25
577,344 7,460,274
5.6699 0.7189
Rule 26
7,128,808 7,128,808
0.7490 0.7516
Rule 27
581,862 583,135
4.7032 4.8014
Rule 28
7,128,808 7,128,808
0.9451 0.9487
Rule 29
656,754 665,263
6.8570 7.0826
Rule 30
8,330,379 8,330,379
0.7445 0.7471
Rule 31
599,846 602,841
6.0157 6.1641
Rule 32
4,725,662 4,725,662
1.0421 1.0472
Rule 33
611,109 611,575
9.1900 9.3105
Rule 34
594,289 4,808,529
11.5468 1.0242
Rule 35
566,893 570,557
6.6264 6.8799

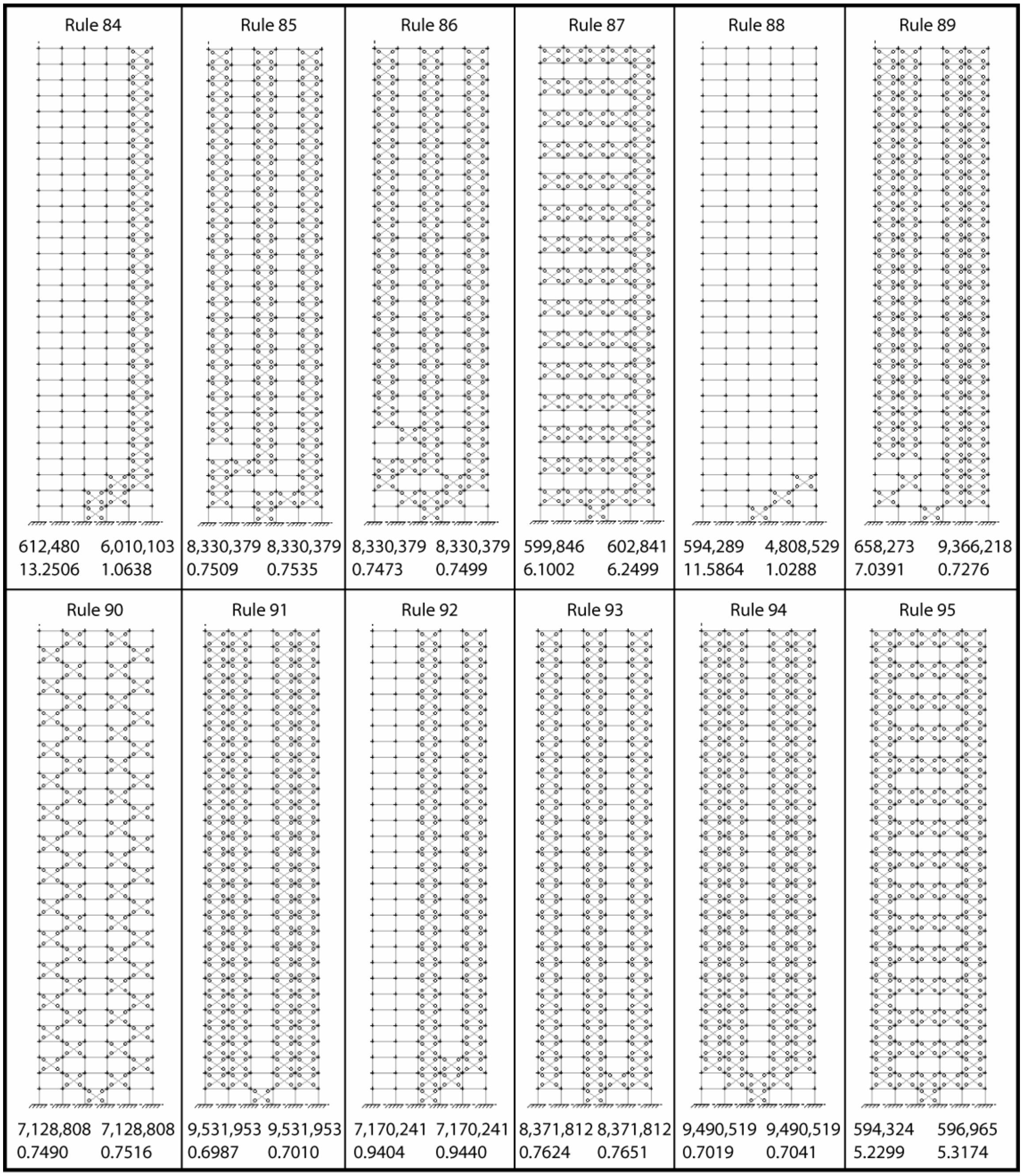
Rule 36	Rule 37	Rule 38	Rule 39	Rule 40	Rule 41
					
600,546 5,927,236 10.9388 0.8976	577,607 580,977 6.2116 6.4089	616,664 5,968,669 13.3142 1.0703	572,386 574,333 6.5672 6.7958	4,725,662 4,725,662 1.0421 1.0472	611,109 611,575 9.1900 9.3105
Rule 42	Rule 43	Rule 44	Rule 45	Rule 46	Rule 47
					
594,289 4,808,529 11.5468 1.0242	572,094 571,063 4.9224 5.0398	600,546 5,927,236 10.9388 0.8976	616,545 628,609 5.2149 10.6965	612,480 6,010,103 13.1778 1.0589	619,478 619,661 4.9025 5.0180

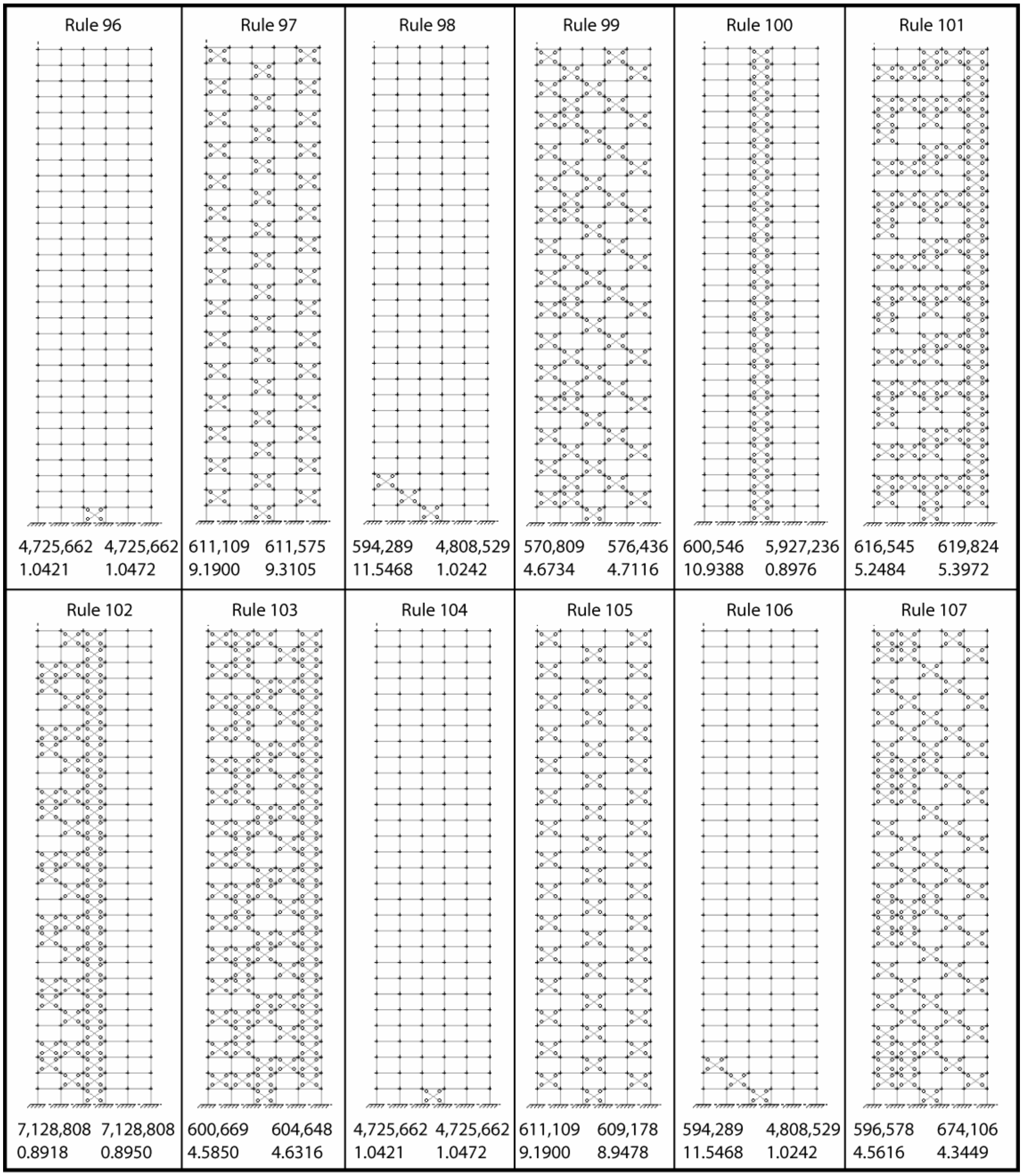


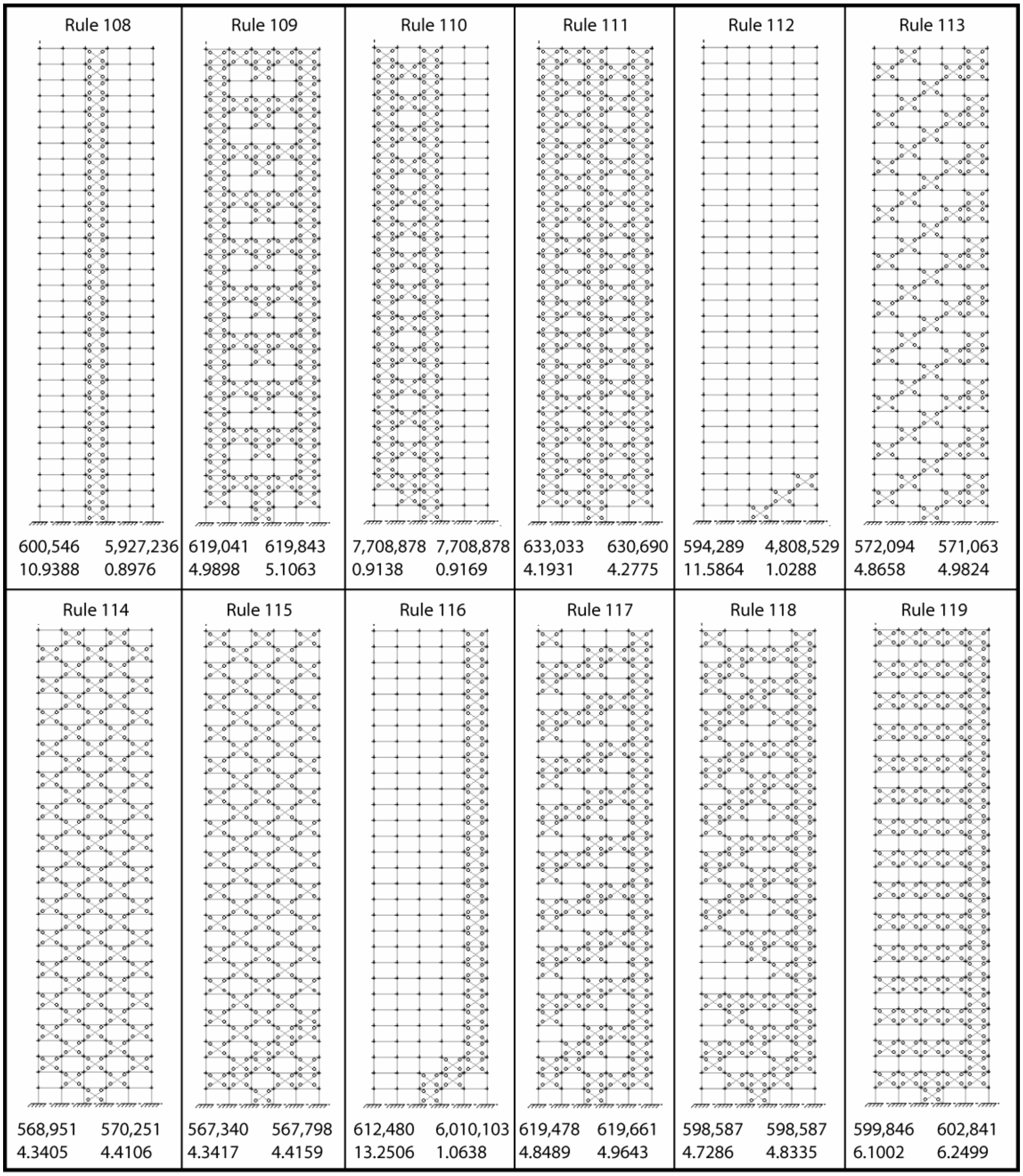
Rule 48
594,289 4,808,529
11.5864 1.0288
Rule 49
566,893 570,557
6.6392 6.8913
Rule 50
568,951 570,251
4.3405 4.4106
Rule 51
560,646 562,570
4.9940 5.1378
Rule 52
616,664 5,968,669
13.3867 1.0752
Rule 53
572,386 574,333
6.5394 6.7675
Rule 54
571,759 571,759
4.6545 4.7564
Rule 55
565,385 569,840
6.6824 6.8606
Rule 56
594,289 4,808,529
11.5864 1.0288
Rule 57
570,809 576,436
4.7417 4.7799
Rule 58
568,951 570,251
4.3405 4.4106
Rule 59
567,340 567,798
4.3478 4.4206

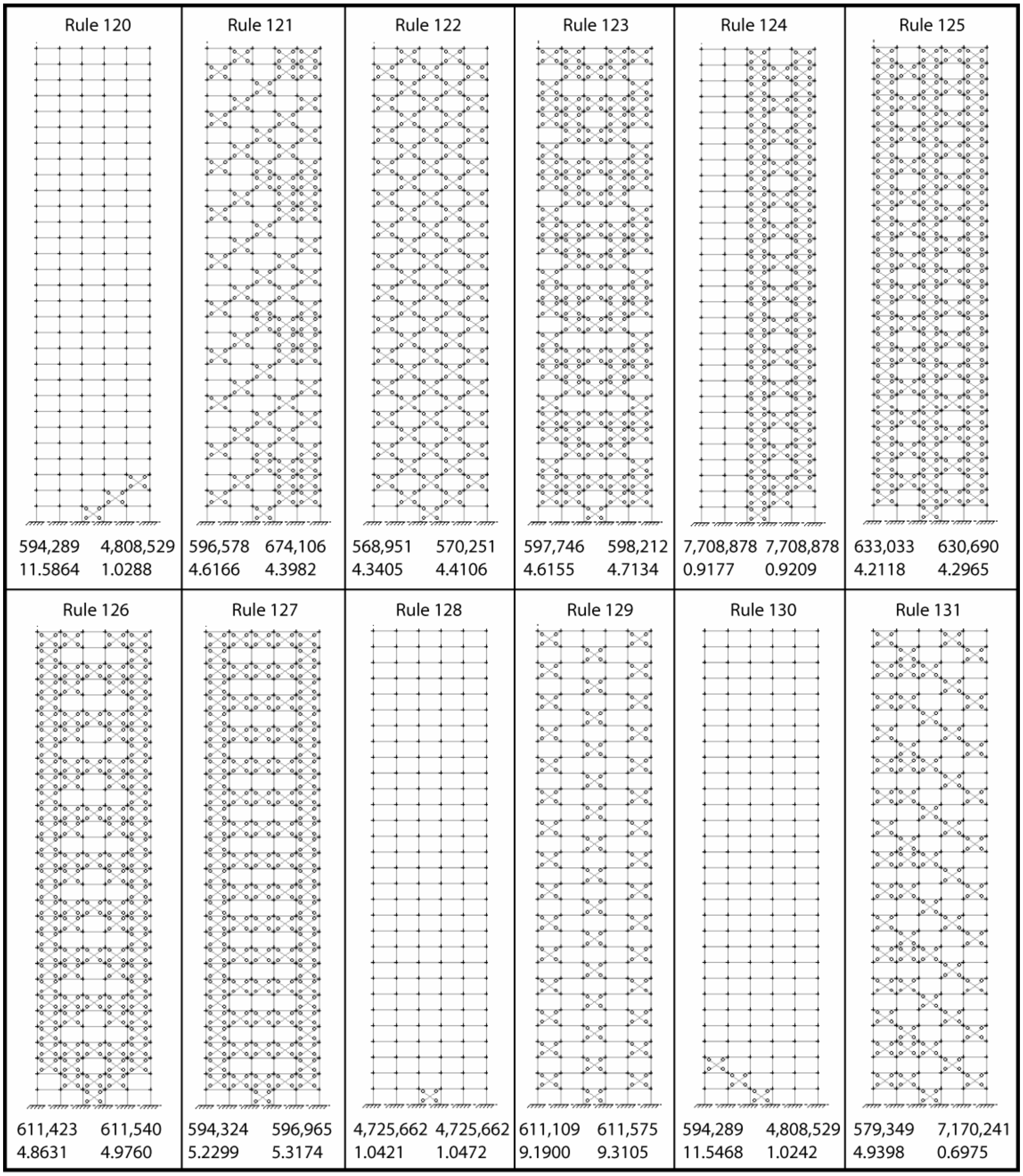

Rule 60
7,128,808 7,128,808
0.8953 0.8984
Rule 61
600,669 601,802
4.5208 4.6089
Rule 62
598,587 598,587
4.6794 4.7843
Rule 63
599,846 602,841
6.0157 6.1641
Rule 64
4,725,662 4,725,662
1.0421 1.0472
Rule 65
611,109 611,575
9.1900 9.3105
Rule 66
594,289 4,808,529
11.5468 1.0242
Rule 67
577,344 7,460,274
5.5718 0.7154
Rule 68
600,546 5,927,236
10.9388 0.8976
Rule 69
8,330,379 8,330,379
0.7645 0.7672
Rule 70
7,128,808 7,128,808
0.9412 0.9448
Rule 71
657,021 8,371,813
7.1272 0.7238

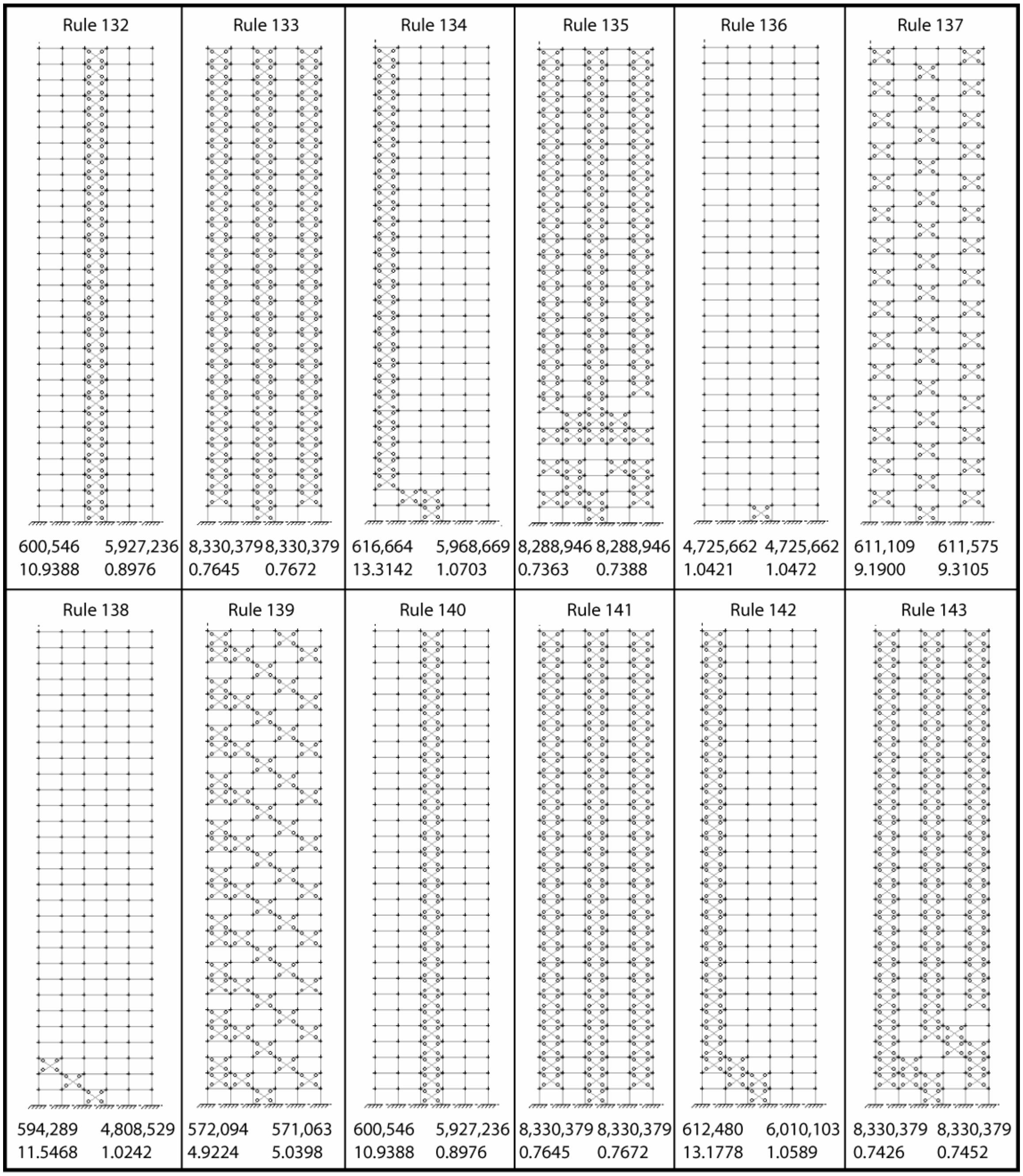

Rule 72
4,725,662 4,725,662
1.0421 1.0472
Rule 73
611,109 611,575
9.1900 9.3105
Rule 74
594,289 4,808,529
11.5468 1.0242
Rule 75
658,273 9,366,218
6.9959 0.7254
Rule 76
600,546 5,927,236
10.9388 0.8976
Rule 77
8,330,379 8,330,379
0.7645 0.7672
Rule 78
7,170,241 7,170,241
0.9365 0.9401
Rule 79
8,371,812 8,371,812
0.7596 0.7622
Rule 80
594,289 4,808,529
11.5864 1.0288
Rule 81
572,094 571,063
4.8658 4.9824
Rule 82
7,128,808 7,128,808
0.7490 0.7516
Rule 83
581,862 583,135
4.7644 4.8629

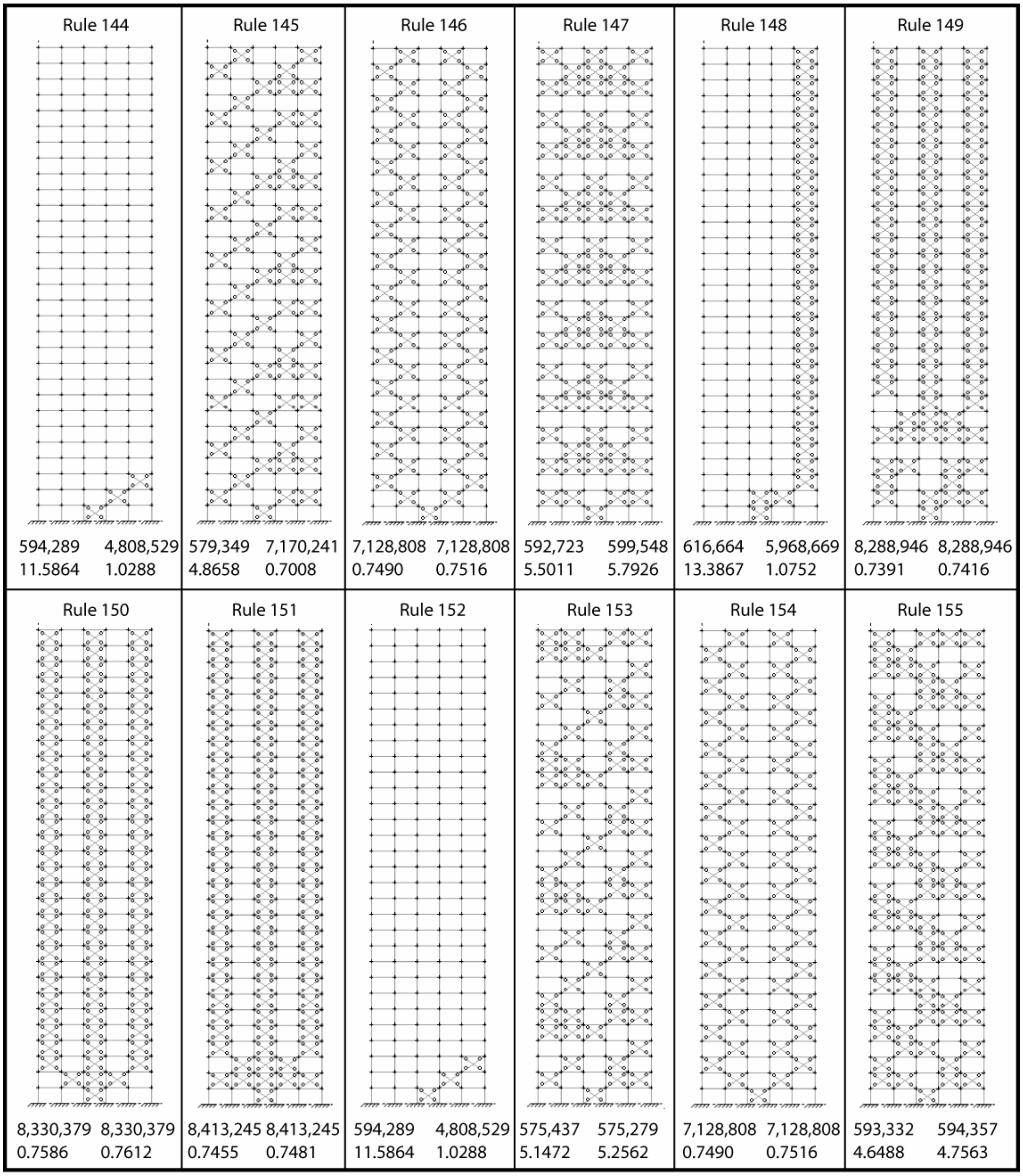

Rule 84
612,480 6,010,103
13.2506 1.0638
Rule 85
8,330,379 8,330,379
0.7509 0.7535
Rule 86
8,330,379 8,330,379
0.7473 0.7499
Rule 87
599,846 602,841
6.1002 6.2499
Rule 88
594,289 4,808,529
11.5864 1.0288
Rule 89
658,273 9,366,218
7.0391 0.7276
Rule 90
7,128,808 7,128,808
0.7490 0.7516
Rule 91
9,531,953 9,531,953
0.6987 0.7010
Rule 92
7,170,241 7,170,241
0.9404 0.9440
Rule 93
8,371,812 8,371,812
0.7624 0.7651
Rule 94
9,490,519 9,490,519
0.7019 0.7041
Rule 95
594,324 596,965
5.2299 5.3174

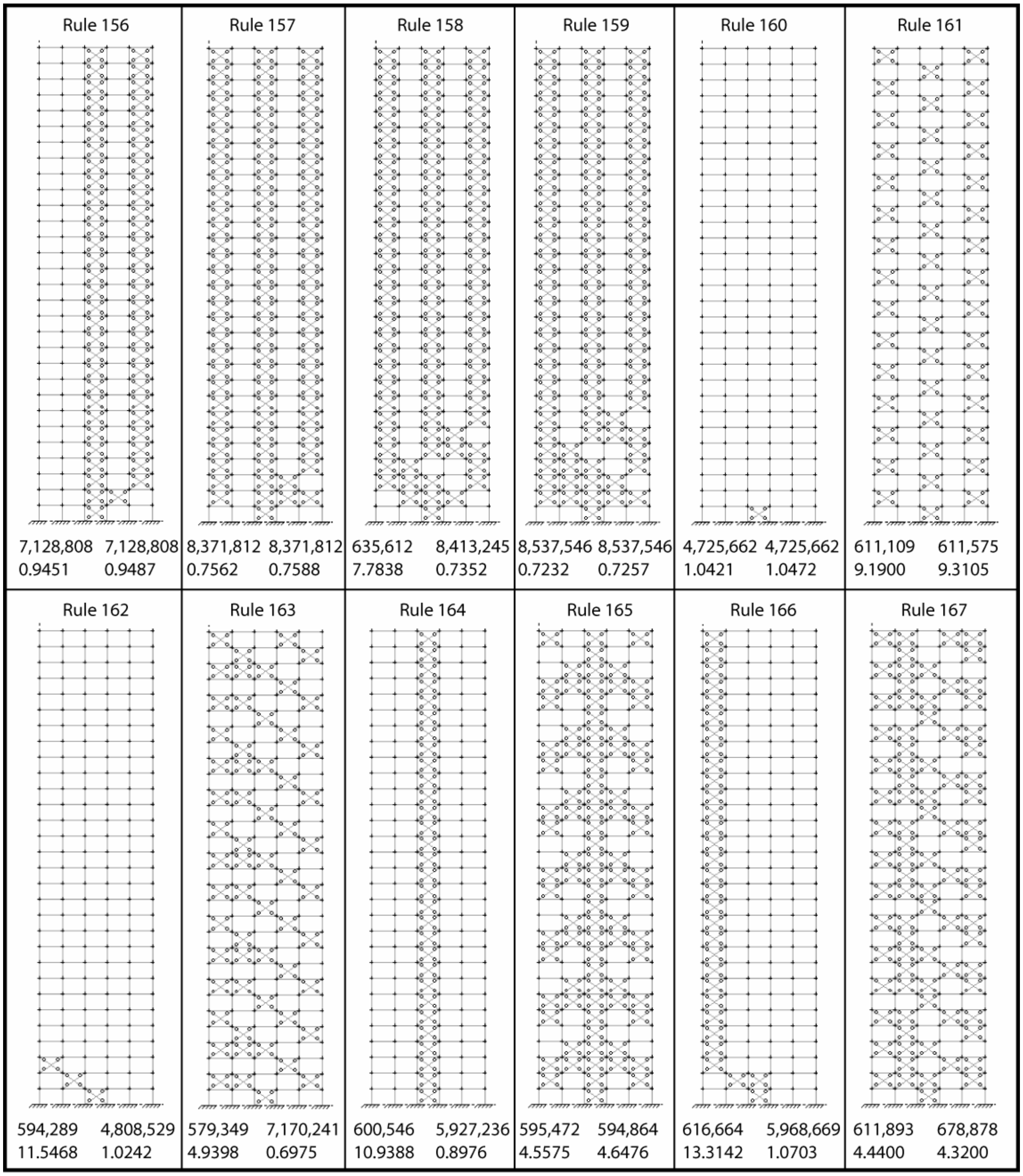

Rule 96
4,725,662 4,725,662
1.0421 1.0472
Rule 97
611,109 611,575
9.1900 9.3105
Rule 98
594,289 4,808,529
11.5468 1.0242
Rule 99
570,809 576,436
4.6734 4.7116
Rule 100
600,546 5,927,236
10.9388 0.8976
Rule 101
616,545 619,824
5.2484 5.3972
Rule 102
7,128,808 7,128,808
0.8918 0.8950
Rule 103
600,669 604,648
4.5850 4.6316
Rule 104
4,725,662 4,725,662
1.0421 1.0472
Rule 105
611,109 609,178
9.1900 8.9478
Rule 106
594,289 4,808,529
11.5468 1.0242
Rule 107
596,578 674,106
4.5616 4.3449

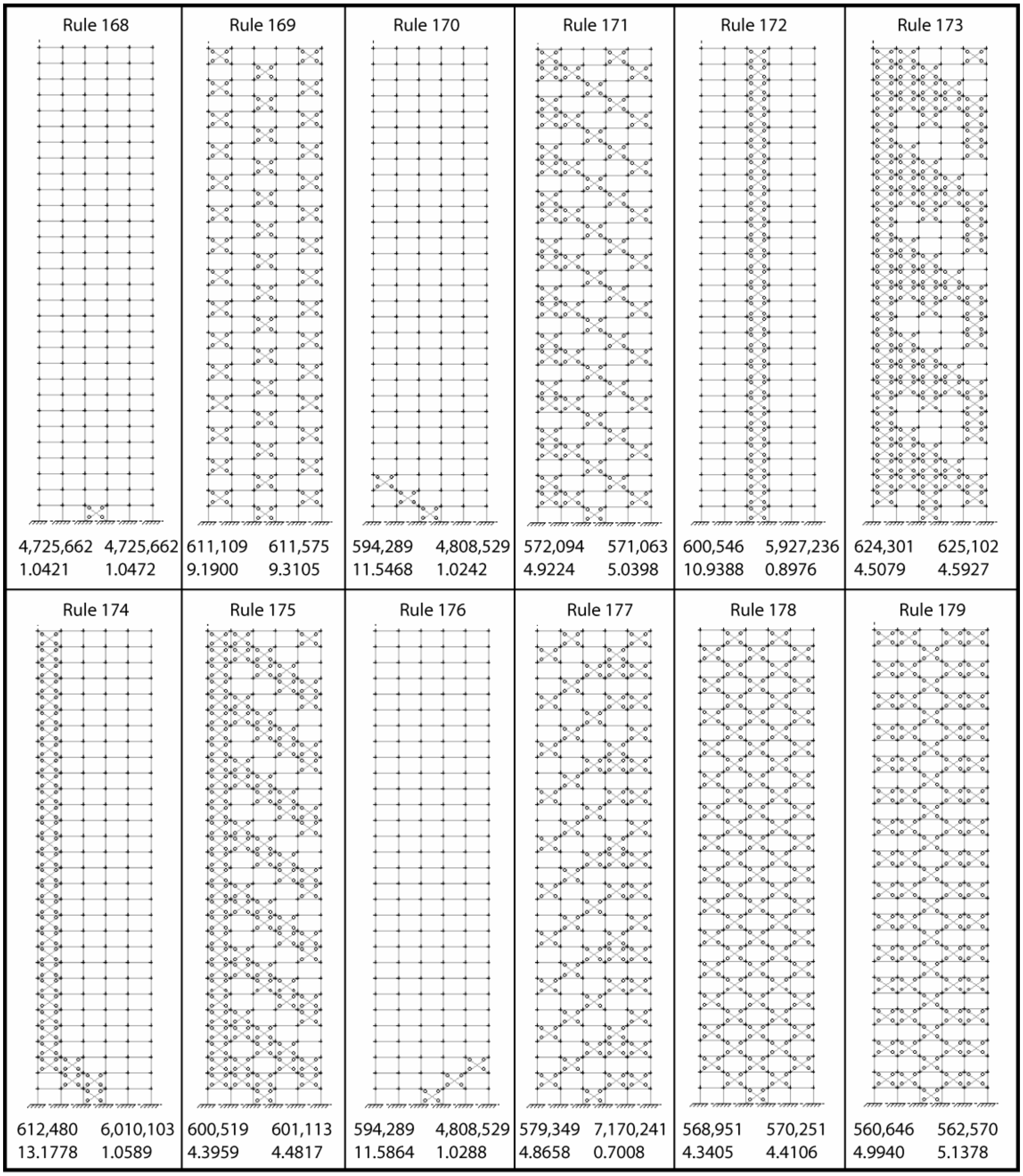

Rule 108
600,546 5,927,236
10.9388 0.8976
Rule 109
619,041 619,843
4.9898 5.1063
Rule 110
7,708,878 7,708,878
0.9138 0.9169
Rule 111
633,033 630,690
4.1931 4.2775
Rule 112
594,289 4,808,529
11.5864 1.0288
Rule 113
572,094 571,063
4.8658 4.9824
Rule 114
568,951 570,251
4.3405 4.4106
Rule 115
567,340 567,798
4.3417 4.4159
Rule 116
612,480 6,010,103
13.2506 1.0638
Rule 117
619,478 619,661
4.8489 4.9643
Rule 118
598,587 598,587
4.7286 4.8335
Rule 119
599,846 602,841
6.1002 6.2499

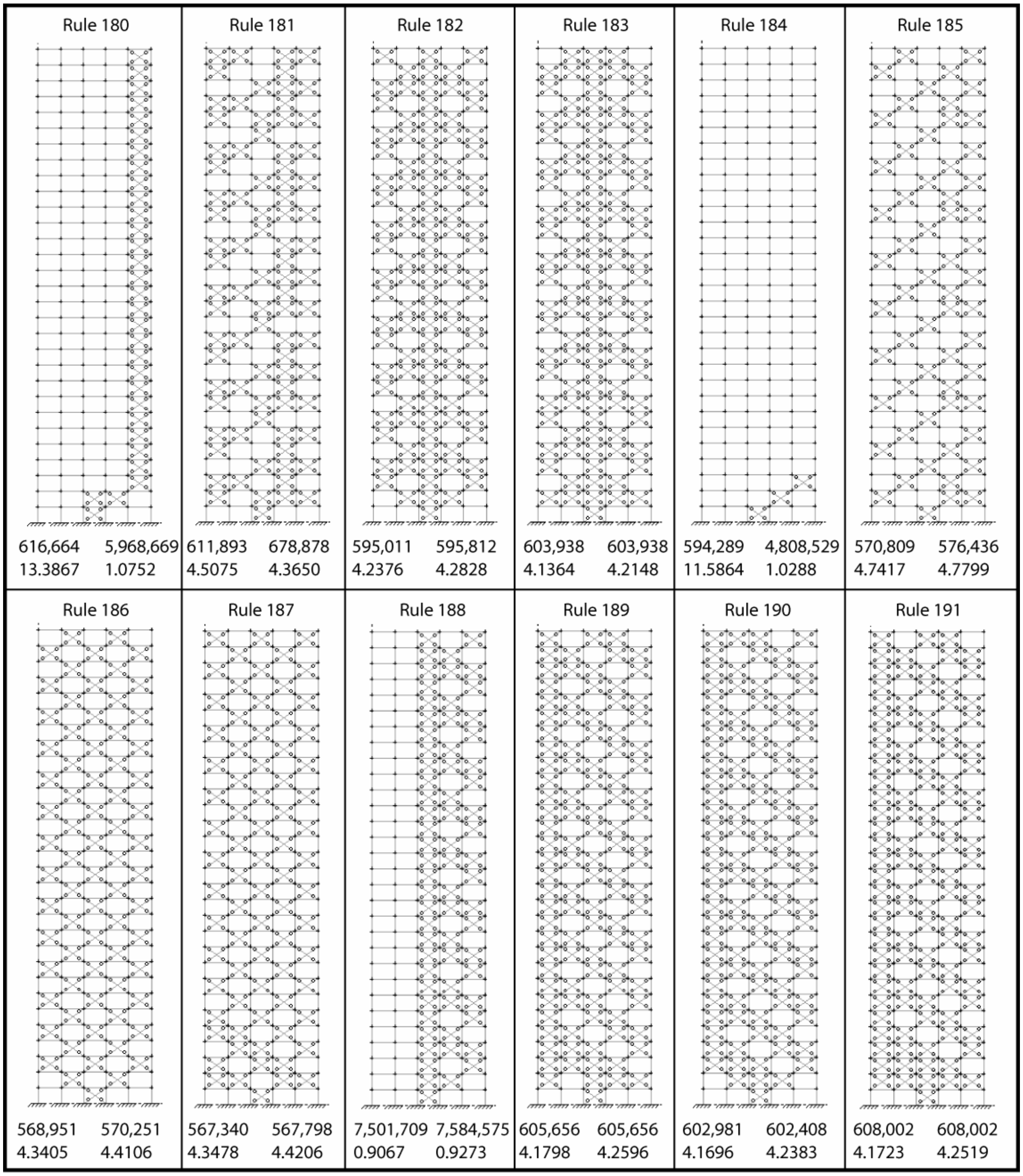

Rule 120
594,289 4,808,529
11.5864 1.0288
Rule 121
596,578 674,106
4.6166 4.3982
Rule 122
568,951 570,251
4.3405 4.4106
Rule 123
597,746 598,212
4.6155 4.7134
Rule 124
7,708,878 7,708,878
0.9177 0.9209
Rule 125
633,033 630,690
4.2118 4.2965
Rule 126
611,423 611,540
4.8631 4.9760
Rule 127
594,324 596,965
5.2299 5.3174
Rule 128
4,725,662 4,725,662
1.0421 1.0472
Rule 129
611,109 611,575
9.1900 9.3105
Rule 130
594,289 4,808,529
11.5468 1.0242
Rule 131
579,349 7,170,241
4.9398 0.6975

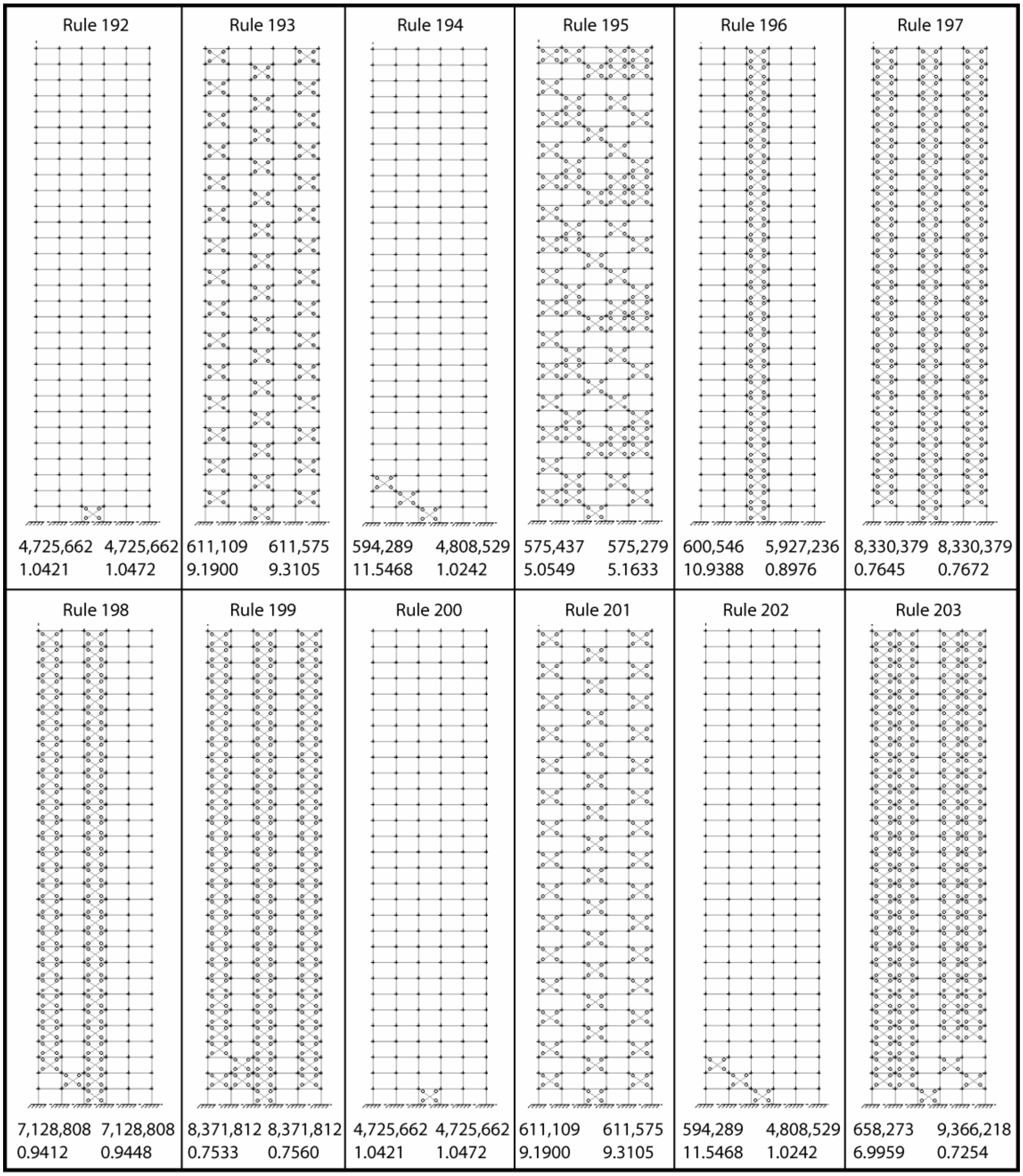

Rule 132
600,546 5,927,236
10.9388 0.8976
Rule 133
8,330,379 8,330,379
0.7645 0.7672
Rule 134
616,664 5,968,669
13.3142 1.0703
Rule 135
8,288,946 8,288,946
0.7363 0.7388
Rule 136
4,725,662 4,725,662
1.0421 1.0472
Rule 137
611,109 611,575
9.1900 9.3105
Rule 138
594,289 4,808,529
11.5468 1.0242
Rule 139
572,094 571,063
4.9224 5.0398
Rule 140
600,546 5,927,236
10.9388 0.8976
Rule 141
8,330,379 8,330,379
0.7645 0.7672
Rule 142
612,480 6,010,103
13.1778 1.0589
Rule 143
8,330,379 8,330,379
0.7426 0.7452

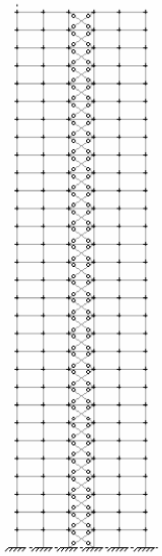
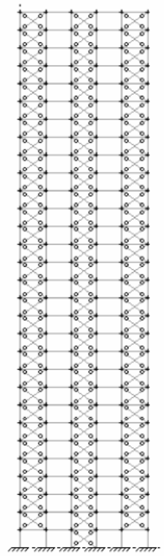
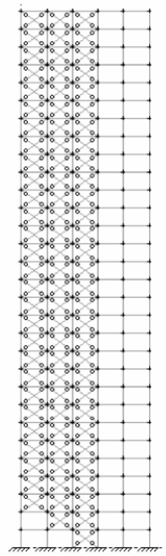
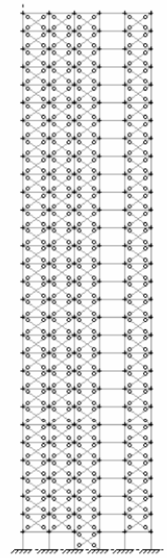
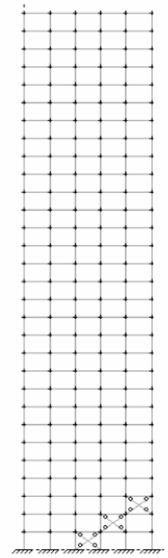
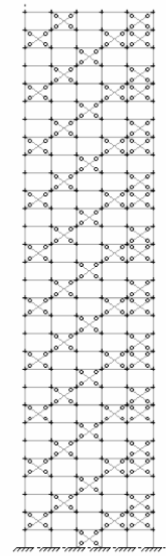

Rule 144
594,289 4,808,529
11.5864 1.0288
Rule 145
579,349 7,170,241
4.8658 0.7008
Rule 146
7,128,808 7,128,808
0.7490 0.7516
Rule 147
592,723 599,548
5.5011 5.7926
Rule 148
616,664 5,968,669
13.3867 1.0752
Rule 149
8,288,946 8,288,946
0.7391 0.7416
Rule 150
8,330,379 8,330,379
0.7586 0.7612
Rule 151
8,413,245 8,413,245
0.7455 0.7481
Rule 152
594,289 4,808,529
11.5864 1.0288
Rule 153
575,437 575,279
5.1472 5.2562
Rule 154
7,128,808 7,128,808
0.7490 0.7516
Rule 155
593,332 594,357
4.6488 4.7563

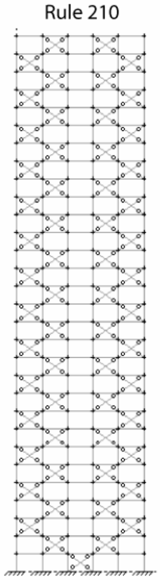
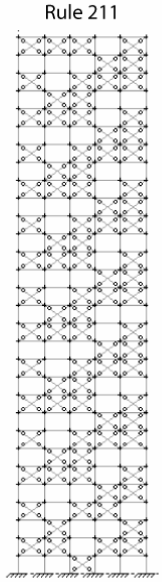
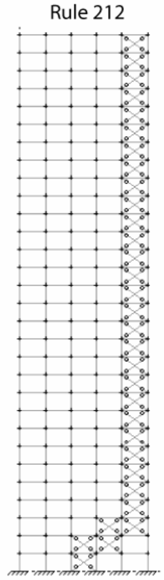
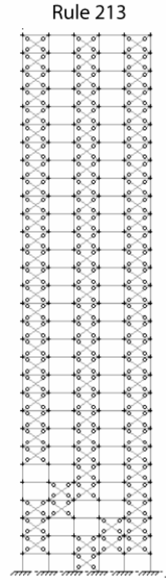
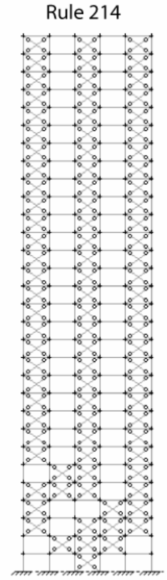
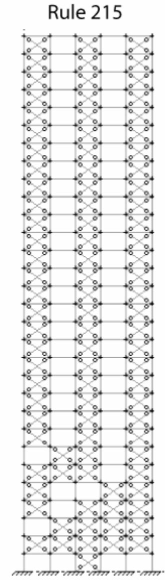

Rule 156
7,128,808 7,128,808
0.9451 0.9487
Rule 157
8,371,812 8,371,812
0.7562 0.7588
Rule 158
635,612 8,413,245
7.7838 0.7352
Rule 159
8,537,546 8,537,546
0.7232 0.7257
Rule 160
4,725,662 4,725,662
1.0421 1.0472
Rule 161
611,109 611,575
9.1900 9.3105
Rule 162
594,289 4,808,529
11.5468 1.0242
Rule 163
579,349 7,170,241
4.9398 0.6975
Rule 164
600,546 5,927,236
10.9388 0.8976
Rule 165
595,472 594,864
4.5575 4.6476
Rule 166
616,664 5,968,669
13.3142 1.0703
Rule 167
611,893 678,878
4.4400 4.3200

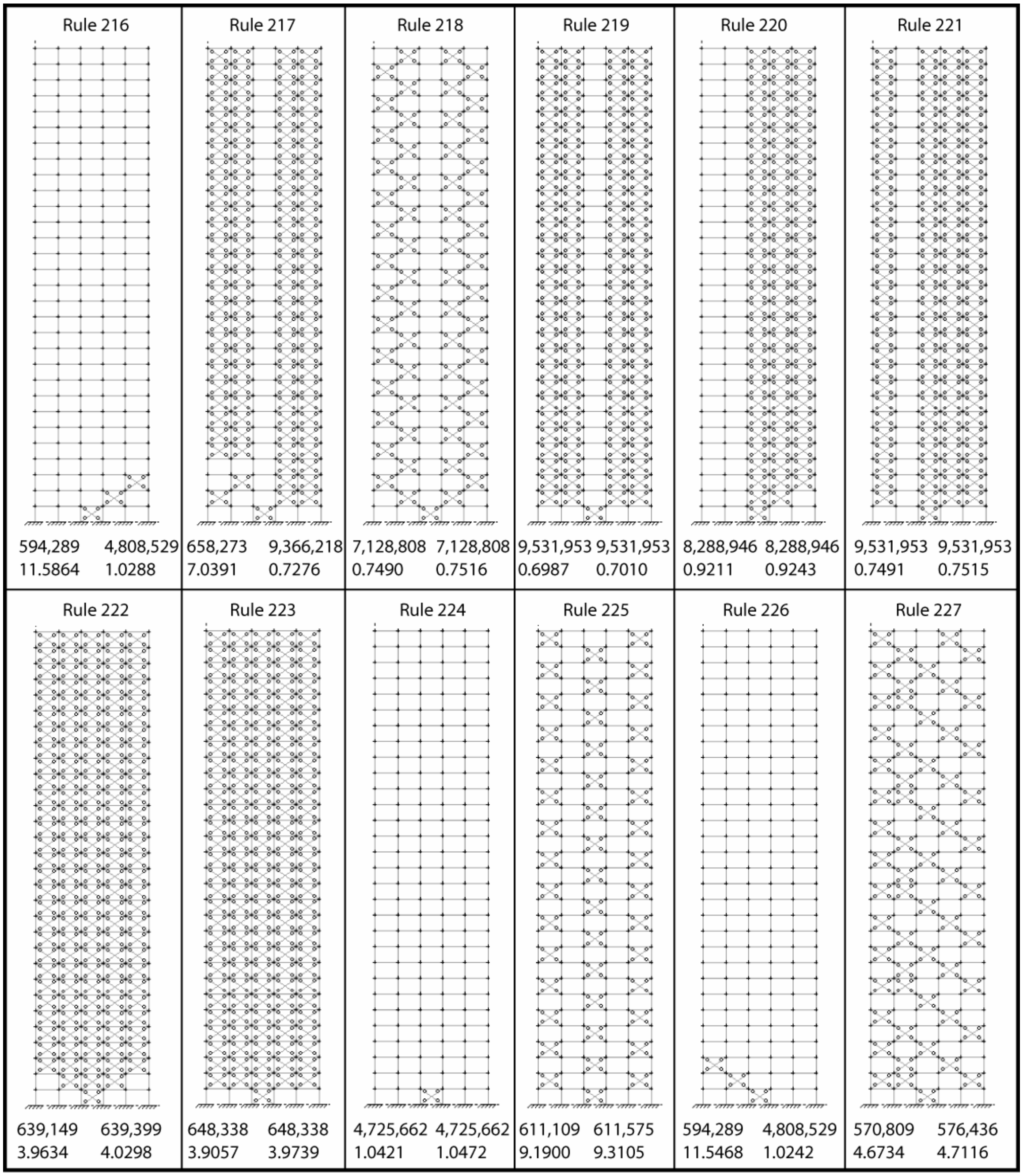

Rule 168
4,725,662 4,725,662
1.0421 1.0472
Rule 169
611,109 611,575
9.1900 9.3105
Rule 170
594,289 4,808,529
11.5468 1.0242
Rule 171
572,094 571,063
4.9224 5.0398
Rule 172
600,546 5,927,236
10.9388 0.8976
Rule 173
624,301 625,102
4.5079 4.5927
Rule 174
612,480 6,010,103
13.1778 1.0589
Rule 175
600,519 601,113
4.3959 4.4817
Rule 176
594,289 4,808,529
11.5864 1.0288
Rule 177
579,349 7,170,241
4.8658 0.7008
Rule 178
568,951 570,251
4.3405 4.4106
Rule 179
560,646 562,570
4.9940 5.1378

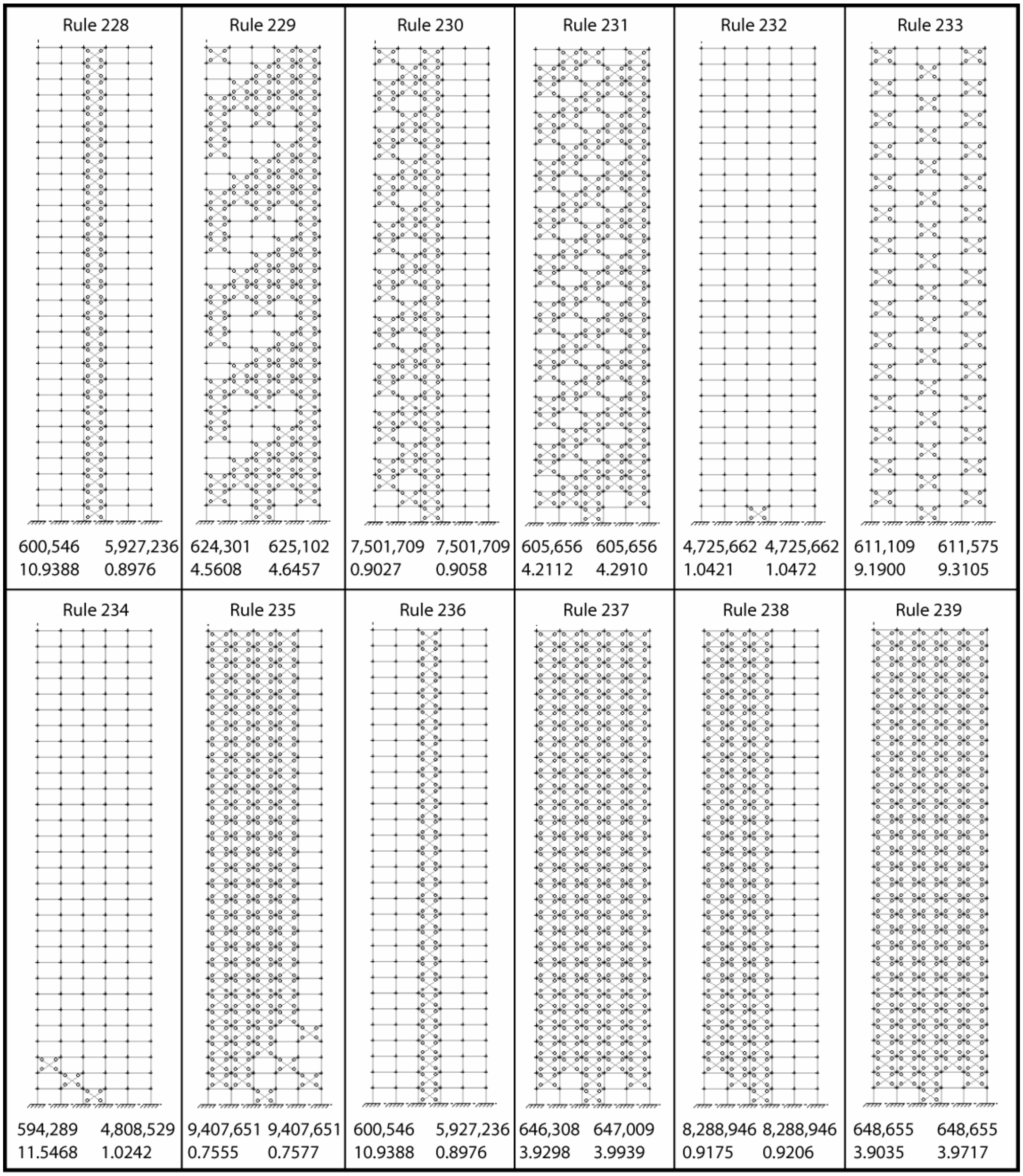

Rule 180
616,664 5,968,669
13.3867 1.0752
Rule 181
611,893 678,878
4.5075 4.3650
Rule 182
595,011 595,812
4.2376 4.2828
Rule 183
603,938 603,938
4.1364 4.2148
Rule 184
594,289 4,808,529
11.5864 1.0288
Rule 185
570,809 576,436
4.7417 4.7799
Rule 186
568,951 570,251
4.3405 4.4106
Rule 187
567,340 567,798
4.3478 4.4206
Rule 188
7,501,709 7,584,575
0.9067 0.9273
Rule 189
605,656 605,656
4.1798 4.2596
Rule 190
602,981 602,408
4.1696 4.2383
Rule 191
608,002 608,002
4.1723 4.2519

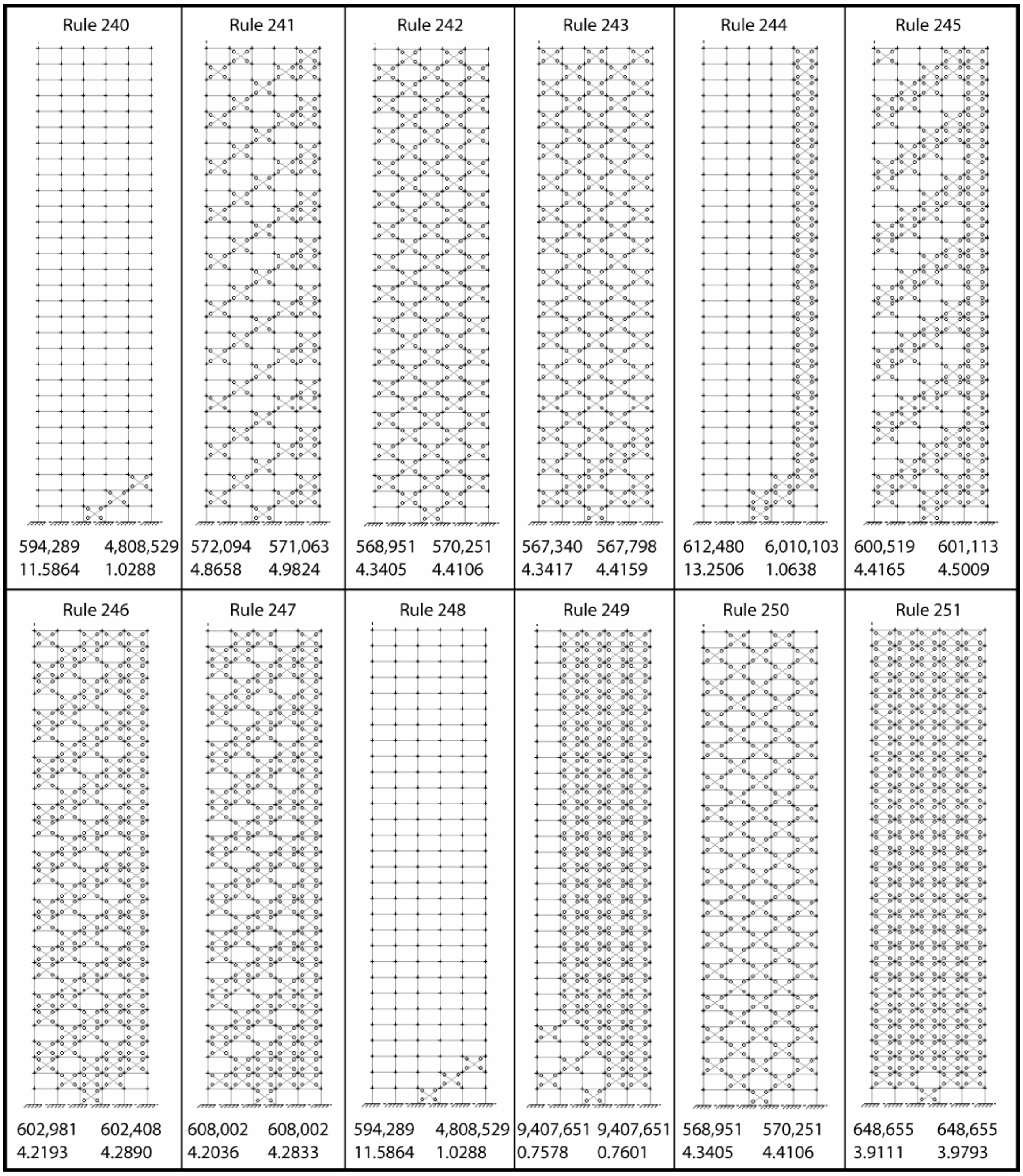

Rule 192
4,725,662 4,725,662
1.0421 1.0472
Rule 193
611,109 611,575
9.1900 9.3105
Rule 194
594,289 4,808,529
11.5468 1.0242
Rule 195
575,437 575,279
5.0549 5.1633
Rule 196
600,546 5,927,236
10.9388 0.8976
Rule 197
8,330,379 8,330,379
0.7645 0.7672
Rule 198
7,128,808 7,128,808
0.9412 0.9448
Rule 199
8,371,812 8,371,812
0.7533 0.7560
Rule 200
4,725,662 4,725,662
1.0421 1.0472
Rule 201
611,109 611,575
9.1900 9.3105
Rule 202
594,289 4,808,529
11.5468 1.0242
Rule 203
658,273 9,366,218
6.9959 0.7254

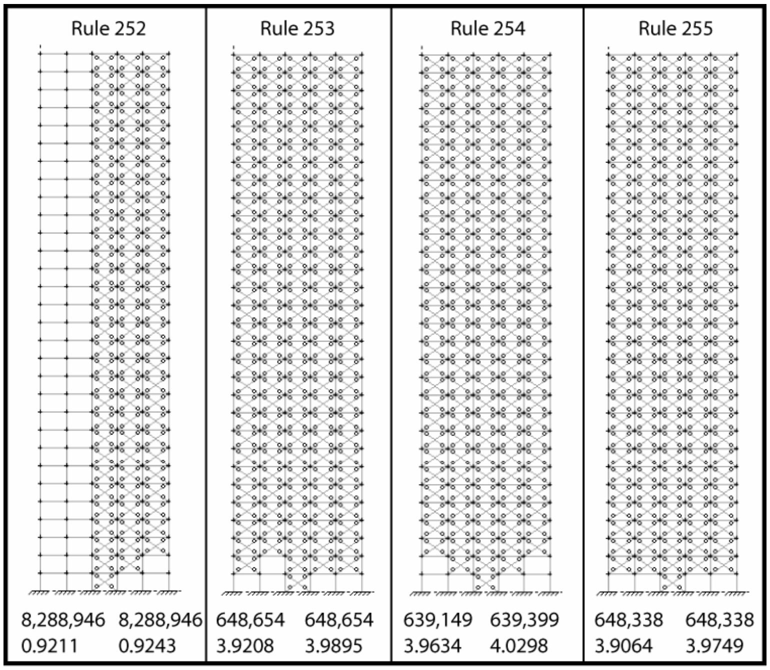
Rule 204	Rule 205	Rule 206	Rule 207	Rule 208	Rule 209
					
600,546 5,927,236	8,330,379 8,330,379	8,288,946 8,288,946	9,531,953 9,531,953	594,289 4,808,529	572,094 571,063
10.9388 0.8976	0.7645 0.7672	0.9175 0.9206	0.7465 0.7489	11.5864 1.0288	4.8658 4.9824

Rule 210	Rule 211	Rule 212	Rule 213	Rule 214	Rule 215
					
7,128,808 7,128,808	593,332 594,357	612,480 6,010,103	8,330,379 8,330,379	635,612 8,413,245	8,537,546 8,537,546
0.7490 0.7516	4.7187 4.8288	13.2506 1.0638	0.7455 0.7480	7.7489 0.7373	0.7261 0.7285


Rule 216
594,289 4,808,529
11.5864 1.0288
Rule 217
658,273 9,366,218
7.0391 0.7276
Rule 218
7,128,808 7,128,808
0.7490 0.7516
Rule 219
9,531,953 9,531,953
0.6987 0.7010
Rule 220
8,288,946 8,288,946
0.9211 0.9243
Rule 221
9,531,953 9,531,953
0.7491 0.7515
Rule 222
639,149 639,399
3.9634 4.0298
Rule 223
648,338 648,338
3.9057 3.9739
Rule 224
4,725,662 4,725,662
1.0421 1.0472
Rule 225
611,109 611,575
9.1900 9.3105
Rule 226
594,289 4,808,529
11.5468 1.0242
Rule 227
570,809 576,436
4.6734 4.7116


Rule 228
600,546 5,927,236
10.9388 0.8976
Rule 229
624,301 625,102
4.5608 4.6457
Rule 230
7,501,709 7,501,709
0.9027 0.9058
Rule 231
605,656 605,656
4.2112 4.2910
Rule 232
4,725,662 4,725,662
1.0421 1.0472
Rule 233
611,109 611,575
9.1900 9.3105
Rule 234
594,289 4,808,529
11.5468 1.0242
Rule 235
9,407,651 9,407,651
0.7555 0.7577
Rule 236
600,546 5,927,236
10.9388 0.8976
Rule 237
646,308 647,009
3.9298 3.9939
Rule 238
8,288,946 8,288,946
0.9175 0.9206
Rule 239
648,655 648,655
3.9035 3.9717


Rule 240
594,289 4,808,529
11.5864 1.0288
Rule 241
572,094 571,063
4.8658 4.9824
Rule 242
568,951 570,251
4.3405 4.4106
Rule 243
567,340 567,798
4.3417 4.4159
Rule 244
612,480 6,010,103
13.2506 1.0638
Rule 245
600,519 601,113
4.4165 4.5009
Rule 246
602,981 602,408
4.2193 4.2890
Rule 247
608,002 608,002
4.2036 4.2833
Rule 248
594,289 4,808,529
11.5864 1.0288
Rule 249
9,407,651 9,407,651
0.7578 0.7601
Rule 250
568,951 570,251
4.3405 4.4106
Rule 251
648,655 648,655
3.9111 3.9793



Rule 252
Rule 253
Rule 254
Rule 255
8,288,946 8,288,946
0.9211 0.9243
648,654 648,654
3.9208 3.9895
639,149 639,399
3.9634 4.0298
648,338 648,338
3.9064 3.9749

REFERENCES

Adeli, H., and Cheng, N. T. (1993). "Integrated genetic algorithm for optimization of space structures." *Journal of Aerospace Engineering*, 6(4), 315-328.

Adeli, H., and Cheng, N. T. (1994). "Augmented Lagrangian genetic algorithm for structural optimization." *Journal of Structural Engineering*, 7(3), 104-118.

Allaire, G., Bonnetier, E., Francfort, G., and Jouve, F. (1997). "Shape optimization by the homogenization method." *Nuemerische Mathematik*, 76, 27-68.

Allaire, G., and Henrot, A. (2001). "On some recent advances in shape optimization." *Comptes Rendus de l Academie des Sciences Series IIB Mechanics*, 329(5), 383-396.

Alligood, K. T., Sauer, T., and Yorke, J. A. (1996). *Chaos: an introduction to dynamical systems*, Springer-Verlag, New York.

Altshuller, G. (1969). *Algorithm of invention*, Moskowskij Raboczij Publishing House, Moscow.

Altshuller, G. (1999). *The innovation algorithm: TRIZ, systematic innovation and technical creativity*, Technical Innovation Center.

American Institute of Steel Construction. (1989). *AISC Manual of Steel Construction*, Ninth edition, AISC, Chicago, IL.

Anagnostou, G., Ronquist, E., and Patera, A. (1992). "A computational procedure for part design." *Computer Methods in Applied Mechanics and Engineering*, 97, 33-48.

Angeline, P. J., and Pollack, J. B. (1993). "Competitive environments evolve better solutions for complex tasks." *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93)*, S. Forrest, ed., Urbana-Champaign, IL, USA, 264–270.

Annicchiarico, W., and Cerrolaza, M. (1999). "Finite elements, genetic algorithms and B-splines: a combined technique for shape optimization." *Finite Elements in Analysis and Design*, 33, 125-141.

Annicchiarico, W., and Cerrolaza, M. (2001). "Structural shape optimization 3D finite-element models based on genetic algorithms and geometric modeling." *Finite Elements in Analysis and Design*, 37(5), 403-415.

Antonsson, E. K., and Cagan, J. (2001). "Formal engineering design synthesis." Cambridge University Press, Cambridge, UK.

Arafat, G. H., Arciszewski, T., and Goodman, B. (1992). "RAMZES: a knowledge-based system for structural concepts evaluation." *Computing Systems in Engineering*, 4(2-3), 211-221.

Arciszewski, T. (1977). "Morphological analysis." Problem, method, solution. Techniques of creative thinking, part I, PWN Publishing House, Warsaw, Poland, 76-96.

Arciszewski, T. (1988). "Stochastic form optimization." Engineering Optimization, Gordon and Breach Publishers, 17-33.

Arciszewski, T., and De Jong, K. A. (2001). "Evolutionary computation in civil engineering: research frontiers." *Proceedings of the Eight International Conference on Civil and Structural Engineering Computing*, B. H. V. Topping, ed., Eisenstadt, Vienna, Austria.

Arciszewski, T., De Jong, K. A., and Vyas, H. (1999). "Inventive design in structural engineering: evolutionary computation approach." *Proceedings of the Fifth International*

Conference on the Applications of AI to Civil and Structural Engineering, B. Kumar and B. H. V. Topping, eds., Oxford, England, 1-9.

Arciszewski, T., and Michalski, R. S. (1984). "Inferential design theory." *Proceedings of the Third International Conference on Artificial Intelligence in Design*, J. S. Gero and F. Sudweeks, eds., Lausanne, Switzerland, 295-309.

Arciszewski, T., Michalski, R. S., and Wnek, J. (1995). "Constructive induction: the key to design creativity." Preprints of the Third International Round-Table Conference on Computational Models of Creative Design, J. S. Gero and M. L. Maher, eds., Heron Island, Queensland, Australia, 397-426.

Arciszewski, T., Sauer, T., and Schum, D. (2003). "Conceptual designing: chaos-based approach." *Journal of Intelligent and Fuzzy Systems*, 13(1/2002/2003), 45-60.

Arora, J. S. (1989). *Introduction to optimum design*, McGraw Hill.

Axelrod, R. M. (1984). *The evolution of cooperation*, Basic Books, New York.

Axelrod, R. M. (1987). "Evolving new strategies: the evolution of strategies in the iterated prisoner's dilemma." Genetic Algorithms and Simulated Annealing, L. Davis, ed., Morgan Kaufmann Publishers, San Mateo, CA, 32-41.

Axtell, R., and Epstein, J. M. (1996). *Growing artificial societies: social science from the bottom*, MIT Press.

Azid, I. A., Kwan, A. S. K., and Seetharamm, K. N. (2002). "An evolutionary approach for layout optimization of a three-dimensional truss." *Structural and Multidisciplinary Optimization*, 24(4), 333–337.

Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*, Oxford University Press, Oxford, New York.

Barlas, Y., and Carpenter, S. (1990). "Philosophical roots of model validation: two paradigms." *System Dynamics Review*, 6(2), 148-166.

Baron, P., Fisher, R., Mill, F., Sherlock, A., and Tuson, A. (1997). "A voxel-based representation for the evolutionary shape optimization of a simplified beam: a case-study of a problem-centered approach to genetic operator design." *Proceedings of the 2nd On-line World Conference on Soft Computing in Engineering Design and Manufacturing (WSC2)*.

Bean, J. C. (1994). "Genetics and random keys for sequencing and optimization." *ORSA Journal on Computing*, 6, 154-160.

Bean, J. C., and Hadj-Alouane, A. B. (1992). "A dual genetic algorithm for bounded integer programs." *TR 92-53*, Department of Industrial and Operations Engineering, University of Michingan.

Bendsoe, M. P., and Kikuchi, N. (1988). "Generating optimal topologies in structural design using a homogenization method." *Computer Methods in Applied Mechanics and Engineering*, 71, 197-224.

Bendsoe, M. P., and Sigmund, O. (2002). *Topology optimization: theory, methods and applications*, Springer-Verlag.

Bennet, J. A., and Botkin, M. E. (1986). "The optimum shape: automated structural design." Plenum Press, New York, London.

Bentley, P. J. (1996). "Generic evolutionary design of solid objects using a genetic algorithm," Ph.D. Dissertation, Division of Computing and Control Systems, Department of Engineering, University of Huddersfield, Queensgate, Huddersfield, UK.

Bentley, P. J. (1999a). "Evolutionary design by computers." Morgan Kaufmann Publishers, San Francisco, CA.

Bentley, P. J. (1999b). "From coffee tables to hospitals: generic evolutionary design." Evolutionary design by computers, P. J. Bentley, ed., Morgan Kaufmann Publishers, San Francisco, CA.

Bentley, P. J. (1999c). "An introduction to evolutionary design by computers." Evolutionary Design by Computers, P. J. Bentley, ed., Morgan Kaufmann Publishers, San Francisco, CA.

Bentley, P. J. (2000). "Exploring component-based representations." *Proceedings of the Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2000)*, I. C. Parmee, ed., University of Plymouth, UK, 161-172.

Bentley, P. J., and Corne, D. W. (2002). "Creative evolutionary systems." Morgan Kaufmann Publishers, San Francisco, CA.

Bentley, P. J., and Kumar, S. (1999). "Three ways to grow designs: a comparison of embryogenies for an evolutionary design problem." *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. J. Jakiela, and R. E. Smith, eds., Orlando, Florida, USA, 35-43.

Berke, L., and Khot, N. S. (1987). "Structural optimization using optimality criteria." Computer Aided Optimal Design: Structural and Mechanical System, C. A. Mota Soares, ed., Springer-Verlag, Berlin, 235-269.

Bilchev, G., and Parmee, I. C. (1995). "The ant colony metaphor for searching continuous design spaces." *Proceedings of the Evolutionary Computing*, T. C. Fogarty, ed., Sheffield, UK, 25-39.

Bilchev, G., and Parmee, I. C. (1996). "Constrained and multi-modal optimisation with an ant colony search model." *Proceedings of the Second International Conference on Adaptive Computing in Engineering Design and Control*, I. C. Parmee and M. J. Denham, eds., Plymouth, UK.

Birkhoff, G. D. (1927). "On the periodic motions of dynamical systems." *Acta Mathematica*, 50, 359-379.

Boden, M. A. (1992). *The creative mind: myths and mechanisms*, Basic Books, New York.

Bohnenberger, O., Hesser, J., and Männer, R. (1995). "Automatic design of truss structures using evolutionary algorithms." *Proceedings of the Second IEEE International Conference on Evolutionary Computation (ICEC'95)*, Perth, Australia, 143-149.

Bonham, C. R., and Parmee, I. C. (2000). "Further developments of the cluster oriented genetic algorithm for design space decomposition." *Submitted to IEEE Transactions in Evolutionary Computation*.

Bramlette, M. F., and Bouchard, E. E. (1991). "Genetic algorithms in parametric design of aircraft." Handbook of Genetic Algorithms, L. Davis, ed., Van Nostrand Reinhold, New York, 109-123.

Camp, C. V., Pezeshk, S., and Cao, G. (1998). "Optimized design of two-dimensional structures using a genetic algorithm." *Journal of Structural Engineering*, 124(5), 551-559.

Campbell, M. I., Cagan, J., and Kotovsky, K. (1999). "A-design: an agent-based approach to conceptual design in dynamic environment." *Research in Engineering Design*, 11(3), 172-192.

Carlson, S. E. (1995). "A general method for handling constraints in genetic algorithms." *Proceedings of the Second Annual Joint Conference on Information Science*, 663-667.

Caroll, C. W. (1961). "The created response surface technique for optimizing nonlinear restrained systems." *Operations Research*, 9, 169-184.

Cartwright, M. L., and Littlewood, J. E. (1951). "Some fixed point theorems." *Annals of Mathematics*, 54, 1-37.

Cerrolaza, M., and Annicchiarico, W. (1999). "Genetic algorithms in shape optimization: finite and boundary element applications." Evolutionary algorithms in engineering and computer science, K. Miettinen, M. M. Makela, P. Neittaanmaki, and J. Periaux, eds., John Wiley & Sons, Chichester, England.

Chan, K. H., Frazer, J. H., and Tang, M.-X. (2000). "Handling the evolution and hierarchy nature of designing in computer-based design support systems." *Proceedings of the Third International Conference on Computer-Aided Industrial Design and Conceptual Design (CAID&CD '2000)*, Hong Kong, 447-454.

Chan, K. H., Frazer, J. H., and Tang, M.-X. (2002). "An evolutionary framework for enhancing design." Artificial Intelligence in Desing '02, J. S. Gero, ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 383-403.

Chankong, V., and Haimes, Y. Y. (1983). *Multiobjective decision making: theory and methodology*, North Holland, New York.

Chapman, C. D., Saitou, K., and Jakiela, M. J. (1993). "Genetic algorithms as an approach to configuration and topology design." *Proceedings of the ASME 19th Design Automation Conference: Advances in Design Automation*, New York, 485-498.

Chapman, C. D., Saitou, K., and Jakiela, M. J. (1994). "Genetic algorithm as an approach to configuration and topology design." *Journal of Mechanical Design*, 116, 1005-1012.

Charnes, A., and Cooper, W. W. (1961). *Management models and industrial applications of linear programming*, John Wiley, New York.

Chawdhry, P., Roy, R., and Pant, R. (1998). "Soft computing in engineering design and manufacturing." 2nd Online World Conference in Soft Computing, Springer, London ; New York, xix, 454.

Chen, S.-Y., and Rajan, S. D. (1998). "Improving the efficiency of genetic algorithms for frame designs." *Engineering Optimization*, 30, 281-307.

Chen, Y. L., and Liu, C. C. (1994). "Multiobjective VAR planning using the goal-attainment method." *IEE Proceedings on Generation, Transmission and Distribution*, 141(3), 227-232.

Cheng, F. Y., and Li, D. (1997). "Multi-objective optimization design with Pareto genetic algorithm." *Journal of Structural Engineering*, 123(9), 1252-1261.

Cheng, R., Gen, M., and Tsujimura, Y. (1996). "A tutorial survey of job-shop scheduling problems using genetic algorithms: I. Representation." *Computers and Industrial Engineering*, 30(4), 983-997.

Chipperfield, A. J., and Fleming, P. J. (1995). "Gas turbine engine controller design using multiobjective genetic algorithms." *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95)*, A. M. S. Zalzala, ed., Halifax Hall, University of Sheffield, UK.

Chung, C.-J., and Reynolds, R. G. (1996). "A testbed for solving optimization problems using cultural algorithms." *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, L. J. Fogel, P. J. Angeline, and T. Bäck, eds., San Diego, CA, USA.

Clarke, D. (2000). "Strategically evolving the future: directed evolution and technological systems development." *Journal of Technological Forecasting and Social Change, Special Issue "Innovation: the Key to Progress in Technology and Society"*, 64(2&3), 133-154.

Coates, P. (1997). "Using genetic programming and L-systems to explore 3D design worlds." *Proceedings of the CAAD Futures '97*, R. Junge, ed., Munich, Germany.

Coello Coello, C. A. (1999). "A comprehensive survey of evolutionary-based multi-objective optimization techniques." *Knowledge and Information Systems*, 1(3), 269-308.

Coello Coello, C. A. (2000a). "Constraint-handling using an evolutionary multiobjective optimization technique." *Civil Engineering and Environmental Systems*, 17, 319-346.

Coello Coello, C. A. (2000b). "Treating constraints as objectives for single-objective evolutionary optimization." *Engineering Optimization*, 32(3), 275-308.

Coello Coello, C. A. (2000c). "An updated survey of GA-based multiobjective optimization techniques." *ACM Computing Surveys*, 32(2), 109-143.

Coello Coello, C. A. (2000d). "Use of a self-adaptive penalty approach for engineering optimization problems." *Computers in Industry*, 41(2), 113-127.

Coello Coello, C. A. (2002). "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art." *Computer Methods in Applied Mechanics and Engineering*, 191, 1245-1287.

Coello Coello, C. A., and Christiansen, A. D. (1998). "Two new GA-based methods for multiobjective optimization." *Civil Engineering Systems*, 15(3), 207-243.

Coello Coello, C. A., and Christiansen, A. D. (2000). "Multiobjective optimization of trusses using genetic algorithms." *Computers & Structures*, 75(6), 647-660.

Coello Coello, C. A., Rudnick, M., and Christiansen, A. D. (1994). "Using genetic algorithms for optimal design of trusses." *Proceedings of the Sixth International Conference on Tools with Artificial Intelligence (ICTAI '94)*, New Orleans, Louisiana, USA, 88-94.

Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*, Kluwer Academic, New York.

Cohoon, J. P., Hegde, S. U., Martin, W. N., and Richards, D. S. (1987). "Punctuated equilibria: a parallel genetic algorithm." *Proceedings of the Second International Conference on Genetic Algorithms (ICGA'87)*, J. J. Grefenstette, ed., Cambridge, MA, USA, 148–154.

Colorni, A., Dorigo, M., and Maniezzo, V. (1991). "Distributed optimization by ant colonies." *Proceedings of the First European Conference on Artificial Life (ECAL'91)*, Paris, France.

Colorni, A., Dorigo, M., and Maniezzo, V. (1992). "An investigation of some properties of the ant algorithm." *Proceedings of the Second International Conference on Parallel Problem Solving from Nature (PPSN-II)*, R. Männer and B. Manderick, eds., Brussels, Belgium, 515-526.

Courant, R. (1943). "Variational methods for the solution of problems of equilibrium and vibrations." *Bulletin of American Mathematical Society*, 49, 1-23.

Cross, N. (1984). "Developments in design methodology." John Wiley, Chichester.

Cross, N. (1989). *Engineering design methods*, John Wiley, Chichester.

Crutchfield, J. P. (1994). "Is anything ever new?: considering emergence." Complexity: Metaphors, Models, and Reality. Santa Fe Institute Studies in the Sciences of Complexity XIX, G. Cowan, D. Pines, and D. Melzer, eds., Addision-Wesley, Reading, MA.

Cvetkovic, D. (2000). "Evolutionary multi-objective decision support systems for conceptual design," Ph.D. Dissertation, School of Computing, University of Plymouth, Plymouth, UK.

Cvetkovic, D., and Parmee, I. C. (1999). "Genetic algorithms based systems for conceptual engineering design." *Proceedings of the 12th International Conference on Engineering Design ICED'99*, U. Lindemann, H. Birkhofer, H. Meerkamm, and S. Vajna, eds., München, Germany, 1035-1038.

Cvetkovic, D., Parmee, I. C., and Webb, E. (1998). "Multi-objective optimisation and preliminary airframe design." The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realisation, I. C. Parmee, ed., Springer-Verlag, Plymouth, UK, 255-267.

D'Ambrosio, J. G., and Birmingham, W. P. (1995). "Preference-directed design." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 9, 219-230.

Darwin, C. (1859). *The origin of species by means of natural selection*, J. Murray, London.

Dasgupta, D., and MacGregor, D. (1991). "A structured genetic algorithm." *IKBS-2-91*, University of Strathclyde, UK.

Dasgupta, D., and Michalewicz, Z. (1997). "Evolutionary algorithms in engineering applications." Springer-Verlag, Berlin, Heidelberg.

Davidor, Y. (1989). "Analogous crossover." *Proceedings of the Third International Conference On Genetic Algorithms (ICGA'89)*, J. D. Schaffer, ed., Fairfax, VA, USA, 98-103.

de Barros Leite, J. P., and Topping, B. H. V. (1998). "Improved genetic operators for structural engineering optimization." *Advances in Engineering Software*, 29(7-9), 529-562.

De Jong, E. D., and Oates, T. (2002). "A coevolutionary approach to representation development." *Proceedings of the ICML-2002 Workshop on Development of Representations*, E. D. De Jong and T. Oates, eds., The University of New South Wales, Sydney, Australia.

De Jong, K. A. (to appear). *Evolutionary computation: a unified approach*, MIT Press, Cambridge, MA.

Deb, K. (1991). "Optimal design of a welded beam via genetic algorithms." *AIAA Journal*, 29, 2013-2015.

Deb, K. (1999). "Evolutionary algorithms for multi-criterion optimization in engineering design." Evolutionary algorithms in engineering and computer science : recent advances in genetic algorithms, evolution strategies, evolutionary programming, genetic programming, and industrial applications, K. Miettinen, M. M. Makela, P. Neittaanmaki, and J. Periaux, eds., John Wiley & Sons, Chichester; New York.

Deb, K. (2000). "An efficient constraint handling method for genetic algorithms." *Computer Methods in Applied Mechanics and Engineering*, 186(2-4), 311-338.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, Chichester, New York.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). "Fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II." *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN-VI)*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo Guervós, and H.-P. Schwefel, eds., Paris, France, 849-858.

Deb, K., and Goel, T. (2001). "A hybrid multi-objective evolutionary approach to engineering shape design." *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'2001)*, E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. W. Corne, eds., Zurich, Switzerland, 385-399.

Deb, K., and Gulati, S. (2001). "Design of truss-structures for minimum weight using genetic algorithms." *Finite Elements in Analysis and Design*, 37(5), 447-465.

Descartes, R. ([1641] 1931). *Meditations on first philosophy. The philosophical works of Rene Descartes*, Cambridge University Press, Cambridge.

d'Humieres, D., and Lallemand, P. (1986). "Lattice gas automata for fluid dynamics." *Physica*, 140A, 326-335.

Dimou, C. K., and Koumousis, V. K. (2003). "Genetic algorithms in competitive environments." *Journal of Computing in Civil Engineering*, 17(3), 142-149.

Dixon, J. R. (1987). "On research methodology towards a scientific theory of design." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1(3).

Dorn, W. C., Gomory, R. E., and Greenberg, H. J. (1964). "Automatic design of optimal structures." *Journal de Mecanique*, 3, 25-52.

Dym, C. L. (1994). *Engineering design: a synthesis of views*, Cambridge University Press, New York, NY.

Dym, C. L., and Little, P. (2000). *Engineering design: a project-based introduction*, John Wiley & Sons, New York.

Eshelman, L. J. (1991). "The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination." *Proceedings of the Second Workshop on Foundations of Genetic Algorithms*, G. J. E. Rawlins, ed., Vail, CO, USA, 265--283.

Fairley, A. (1991). "Comparison of methods of choosing the crossover point in the genetic crossover operation." University of Liverpool, Liverpool, UK.

Fermen-Coker, M., and Johnson, G. E. (2000). "Geometric considerations for the prevention of chaos in heated plate vibrations." *International Journal of Bifurcation and Chaos*, 10(6), 1383-1405.

Fiacco, A. V., and McCormick, G. P. (1968). "Extensions to SUMT for nonlinear programming: equality constraints and extrapolation." *Management Science*, 12(11), 816-828.

Finger, S., and Dixon, J. R. (1989a). "A review of research in mechanical engineering design. Part I: descriptive, prescriptive and computer-based models of design processes." *Research in Engineering Design*, 1(1), 51–67.

Finger, S., and Dixon, J. R. (1989b). "A review of research in mechanical engineering design. Part II: representations, analysis and design for the life cycle." *Research in Engineering Design*, 1(2), 121–137.

Fogarty, T. C. (1989). "Varying the probability of mutation in genetic algorithm." *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, J. D. Schaffer, ed., Fairfax, VA, USA, 104-109.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial Intelligence through simulated evolution*, John Wiley, Chichester, UK.

Fonseca, C. M., and Fleming, P. J. (1993). "Genetic algorithms for multi-objective optimization: formulation, discussion, and generalization." *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93)*, S. Forrest, ed., Urbana-Champaign, IL, USA, 416-423.

Forrest, S., and Perelson, A. S. (1990). "Genetic algorithms and the immune system." *Proceedings of the First International Conference on Parallel Problem Solving from Nature (PPSN-I)*, H.-P. Schwefel and R. Männer, eds., Dortmund, Germany, 320-325.

Franceschetti, D. R., Campbell, B. W., and Hamneken, J. W. (1992). "Hamming nets, Ising sets, cellular automata, neural nets and random walks." *American Journal of Physics*, 61, 50-53.

Frazer, J. (1995). *An evolutionary architecture*, Architectural Association Publications, London.

French, M. (1992). *Form, structure and mechanism*, MacMillian, London.

French, M. (1994). *Invention and evolution: design in nature and engineering*, Cambridge University Press, Cambridge, UK.

Funes, P., and Pollack, J. B. (1999). "Computer evolution of buildable objects." Evolutionary design by computers, P. J. Bentley, ed., Morgan Kaufmann Publishers, San Francisco, CA.

Gen, M., and Cheng, R. (1997). *Genetic algorithms and engineering design*, Wiley, New York.

Gen, M., and Cheng, R. (2000). *Genetic algorithms and engineering optimization*, Wiley, New York.

Gerhadrt, M., and Schuster, H. (1989). "A cellular automaton describing the formation of spatially ordered structures in chemical systems." *Physica*, 36D, 209-221.

Gero, J. S. (1992). "Creativity, emergence and evolution in design." *Preprints Computational Models of Creative Design*, 1-28.

Gero, J. S. (1996). "Computers and creative design." The Global Design Studio, M. Tan and R. Teh, eds., National University of Singapore, 11-19.

Gero, J. S. (2002). "Artificial Intelligence in design '02." Kluwer Academic Publishers, Dordrecht, The Netherlands.

Gero, J. S., and Brazier, F. M. (2002). "Agents in design." Key Centre of Design Computing and Cognition, Sydney, Australia.

Gero, J. S., and Kannengiesser, U. (2003). "A function-behavior-structure view of social situated design agents." *Proceedings of the CAADRIA03*.

Gero, J. S., and Schnier, T. (1995). "Evolving representations of design cases and their use in creative design." Preprints Computational Models of Creative Design, J. S. Gero, M. L. Maher, and F. Sudweeks, eds., Key Center of Design Computing, University of Sydney, Syndey, Australia, 343-368.

Goldberg, D. E. (1987). "Computer-aided gas pipeline operation using genetic algorithms and rule learning, part I: Genetic algorithm in pipeline optimization." *Engineering with Computers*, 47-58.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley Pub. Co., Reading, Mass.

Goldberg, D. E., Korb, B., and Deb, K. (1989). "Messy genetic algorithms: motivation, analysis, and first results." *Complex Systems*, 3(5), 493-530.

Goldberg, D. E., and Samtani, M. (1986). "Engineering optimization via genetic algorithm." *Proceedings of the Ninth Conference on Electronic Computation*, University of Alabama, Birmingham, 471-482.

Grabska, E. (1993). "Graphs and designing." *Proceedings of the International Workshop on Graph Transformations in Computer Science*, H. J. Schneider and H. Ehrig, eds., Dagstuhl Castle, Germany, 188-202.

Grefenstette, J. J., and Baker, J. E. (1989). "How genetic algorithms work: a critical look at implicit parallelism." *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, J. D. Schaffer, ed., Fairfax, VA, USA, 20-27.

Greiner, D., Winter, G., and Emperador, J. M. (2001). "Optimizing frame structures by different strategies of genetic algorithms." *Finite Elements in Analysis and Design*, 37, 381-402.

Grierson, D. E. (1989). "Computer-automated optimal design for structural steel frameworks." *Proceedings of the NATO ASI Conference on Optimization and Decision Support Systems in Civil Engineering*, B. H. V. Topping, ed., Edinburgh, UK, 327-354.

Grierson, D. E., and Khajehpour, S. (2002). "Method for conceptual design applied to office buildings." *Journal of Computing in Civil Engineering*, 16(2), 83-103.

Grierson, D. E., and Pak, W. (1993a). "Discrete optimal design using a genetic algorithm." Topology Design of Structures, M. P. Bendsoe and C. A. Mota Soares, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 89-102.

Grierson, D. E., and Pak, W. (1993b). "Optimal sizing, geometrical and topological design using a genetic algorithm." *Structural Optimization*, 6, 151-159.

Hadj-Alouane, A. B., and Bean, J. C. (1997). "A genetic algorithm for the multiple-choice integer program." *Operations Research*, 45, 92-101.

Haimes, Y. Y., Lasdon, L. S., and Wismer, D. A. (1971). "On a bicriterion formulation of the problems of integrated system identification and system optimization." *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3), 296-297.

Hajela, P. (1990). "Genetic search - an approach to the nonconvex optimization problem." *AIAA Journal*, 26, 1205-1212.

Hajela, P. (1992). "Genetic algorithms in automated structural synthesis." Optimization and Artificial Intelligence in Civil and Structural Engineering, B. H. V. Topping, ed., Kluwer Academic Press.

Hajela, P., and Kim, B. (1999). "GA based learning in cellular automata models for structural analysis." *Proceedings of the 3rd World Congress on Structural and Multidisciplinary Optimization*, Niagara Falls, NY.

Hajela, P., and Kim, B. (2001). "On the use of energy minimization for CA based analysis in elasticity." *Structural and Multidisciplinary Optimization*, 23(1), 24-33.

Hajela, P., and Lee, E. (1995a). "Genetic algorithms in truss topological optimization." *Journal of Solids and Structures*, 32(22), 3341-3357.

Hajela, P., Lee, E., and Cho, H. K. (1998). "Genetic algorithms in topologic design of grillage structures." *Computer-Aided Civil and Infrastructure Engineering*, 13(1), 13-22.

Hajela, P., Lee, E., and Lin, C.-Y. (1993). "Genetic algorithms in structural topology optimization." Topology Design of Structures, M. P. Bendsoe and C. A. Mota Soares, eds., 117-133.

Hajela, P., and Lee, J. (1995b). "Constrained genetic search via schema adaptation. An immune network solution." *Proceedings of the First World Congress of Structural and Multidisciplinary Optimization (WCSMO-1)*, N. Olhoff and G. I. N. Rozvany, eds., Goslar, Germany, 915-920.

Hajela, P., and Lee, J. (1996). "Constrained genetic search via schema adaptation. An immune network solution." *Structural Optimization*, 12(1), 11-15.
Hajela, P., and Lin, C.-Y. (1992). "Genetic search strategies in multicriterion optimal design." *Structural Optimization*(4), 99-107.
Hajela, P., and Vittal, S. (2000). "Evolutionary computing and topology optimization: a state of the art assessment." *Proceedings of the NATO Advanced Research Workshop on Topology Optimization*, Budapest, Hungary.
Hall II, E. K., and Hanagud, S. V. (1991). "Active control of chaotic systems by using adaptive structural concepts." *91-1062-CP*, AIAA.
Hamda, H., Jouve, F., Lutton, E., Schoenauer, M., and Sebag, M. (2002a). "Compact unstructured representations for evolutionary topological optimum design." *Applied Intelligence*, 16, 139-155.
Hamda, H., Roudenko, O., and Schoenauer, M. (2002b). "Multi-objective evolutionary topological optimum design." *Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, I. C. Parmee, ed., University of Exeter, Devon, UK, 121-132.
Haslinger, J., and Neittaanmaki, P. (1996). *Finite element approximation for optimal shape design material and topology design*, John Wiley & Sons, Chichester, UK.
Hegel, G. W. F. ([1817] 1959). *Encyclopedia of philosophy*, Philosophical Library, New York.
Hillis, W. D. (1991). "Co-evolving parasites improve simulated evolution as an optimization procedure." Artificial Life II, C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, eds., Addison-Wesley, Santa Fe Institute Studies in the Sciences of Complexity, New Mexico, USA, 313-324.
Hoeffler, A., Leysner, U., and Weidermann, J. (1973). "Optimization of the layout of trusses combining strategies based on Mitchel's theorem and on biological principles of evolution." *Proceedings of the 2nd Symposium on Structural Optimization*, Milan, Italy.
Holland, J. H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, Michigan.
Holmes, P., Domokos, G., and Hek, G. (2000). "Euler buckling in a potential field." *Journal of Nonlinear Science*, 10, 477-505.
Holmes, P., Domokos, G., Schmitt, J., and Szeberenyi, I. (1999). "Constrained Euler buckling: an interplay of computation and analysis." *Computer Methods in Applied Mechanics and Engineering*, 170, 175-207.
Horn, J., and Nafpliotis, N. (1993). "Multiobjective optimization using the niched Pareto genetic algorithm." *IlliGAl Report 93005*, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA.
Hornby, G. S. (2003). "Generative representations for evolutionary design automation," Ph.D. Dissertation, Department of Computer Science, Brandeis University, Waltham, MA.
Ilachinski, A. (2001). *Cellular automata: a discrete universe*, World Scientific, Singapore; River Edge, NJ.
Inou, N., Shimotai, N., and Uesugi, T. (1994). "Cellular automaton generating topological structures." *Proceedings of the Second European Conference on Smart Structures and Materials*, A. McDonach, P. T. Gardiner, R. S. McEwen, and B. Culshaw, eds., Glasgow, UK, 47-50.

Inou, N., Uesugi, T., Iwasaki, A., and Ujihashi, S. (1998). "Self-organization of mechanical structure by cellular automata." Fracture and strength of solids. Part 2: Behavior of materials and structure, P. Tong, T. Y. Zhang, and J. Kim, eds., 1115-1120.

Jacob, C. (1994). "Genetic L-system programming." *Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN-III)*, Y. Davidor, H.-P. Schwefel, and R. Männer, eds., Jerusalem, Israel, 334-343.

Jakiela, M. J., Chapman, C. D., Duda, J., Adewuya, A., and Saitou, K. (2000). "Continuum structural topology design with genetic algorithms." *Computer Methods in Applied Mechanics and Engineering*, 186, 339-356.

Jakob, W., Gorges-Schleuter, M., and Blume, C. (1992). "Application of genetic algorithms to task planning and learning." *Proceedings of the Second International Conference on Parallel Problem Solving from Nature (PPSN-II)*, R. Männer and B. Manderick, eds., Brussels, Belgium, 293-302.

Jarmai, K., Snyman, J. A., Farkas, J., and Gondos, G. (2003). "Optimal design of a welded I-section frame using four conceptually different optimization algorithms." *Structural and Multidisciplinary Optimization*, 25, 54–61.

Jenkins, W. M. (1991a). "Structural optimization with the genetic algorithm." *Structural Engineer*, 69(24), 418-422.

Jenkins, W. M. (1991b). "Towards structural optimization via the genetic algorithm." *Computers & Structures*, 40(5), 1321-1327.

Jenkins, W. M. (1992). "Plane frame optimum design environment based on genetic algorithm." *Journal of Structural Engineering*, 118(11), 3103-3112.

Jenkins, W. M. (1997). "On the application of natural algorithms to structural design optimization." *Engineering structures*, 19(4), 302-308.

Jensen, E. D. (1992). "Topological structural design using genetic algorithms," Ph.D. Dissertation, Purdue University, Lafayette, IN.

Johnson, N. J. (1978). "Modified t tests and confidence intervals for asymmetrical populations." *Journal of the American Statistical Association*, 73(363), 536-544.

Joines, J. A., and Houck, C. R. (1994). "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's." *Proceedings of the First IEEE International Conference on Evolutionary Computation (ICEC'94)*, Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, eds., Orlando, FL, USA, 579-584.

Kane, C., and Schoenauer, M. (1995). "Genetic operators for two-dimensional shape optimization." Artificial Evolution, J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, eds., Springer Verlag.

Kane, C., and Schoenauer, M. (1996). "Topological optimum design using genetic algorithms." *Control and Cybernetics*, 25(5), 1059-1088.

Kant, I. ([1781] 1933). *The critique of pure reason*, St. Martins Press, London.

Kawohl, B., Pironneau, O., Tartar, L., and Zolesio, J.-P. (2000). "Optimal shape design." Lecture Notes in Mathematics vol. 1740, Springer-Verlag, Berlin New York Heidelberg.

Keane, A. J. (1994). "Experiences with optimizers in structural design." *Proceedings of the First International Conference on Adaptive Computing in Engineering Design and Control*, I. C. Parmee, ed., Plymouth, UK, 14-27.

Keane, A. J., and Brown, S. M. (1996). "The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques." *Proceedings of the*

Conference on Adaptive Computing in Engineering Design and Control 96, I. C. Parmee, ed., Plymouth, UK, 107-113.

Kessler, D. A., Levine, H., and Reynolds, W. N. (1990). "Coupled-map lattice model for crystal growth." *Physics Review*, 42A(6125).

Kicinger, R. (2004). "Emergent Engineering Design: Design Creativity and Optimality Inspired by Nature." Adobe Illustrator.

Kicinger, R., and Arciszewski, T. (2004). "Multiobjective evolutionary design of steel structures in tall buildings." *Proceedings of the AIAA 1st Intelligent Systems Technical Conference*, S. Obayashi, ed., Chicago, Illinois.

Kicinger, R., Arciszewski, T., and De Jong, K. A. (2003). "Evolutionary designing of steel structures in tall buildings." *Journal of Computing in Civil Engineering*(tentatively approved).

Kicinger, R., Arciszewski, T., and De Jong, K. A. (2004a). "Distributed evolutionary design: island-model based optimization of steel skeleton structures in tall buildings." *Proceedings of the 10th International Conference on Computing in Civil and Building Engineering (ICCCBE-X)*, K. Beucke, B. Firmenich, D. Donath, R. Fruchter, and K. Roddis, eds., Weimar, Germany, 190.

Kicinger, R., Arciszewski, T., and De Jong, K. A. (2004b). "Evolutionary computation and structural design: a survey of the state of the art." *Computers & Structures*, submitted.

Kicinger, R., Arciszewski, T., and De Jong, K. A. (2004c). "Morphogenesis and structural design: cellular automata representations of steel structures in tall buildings." *Proceedings of the Congress on Evolutionary Computation (CEC'2004)*, Portland, Oregon, 411-418.

Kicinger, R., Arciszewski, T., and De Jong, K. A. (2004d). "Morphogenic evolutionary design: cellular automata representations in topological structural design." Adaptive Computing in Design and Manufacture VI, I. C. Parmee, ed., Springer-Verlag, London, UK, 25-38.

Kicinger, R., De Jong, K. A., and Arciszewski, T. (2002). "Long term versus short term evolutionary design." *Proceedings of the 9th International Workshop of the European Group for Intelligent Computing in Engineering*, Darmstadt, Germany, 184-195.

Kim, D. G. (1998). "Riemann mapping based constraint handling for evolutionary search." *Proceedings of the 1998 ACM Symposium on Applied Computing*, Atlanta, GA, USA, 379-385.

Kim, J.-H., and Myung, H. (1997). "Evolutionary programming techniques for constrained optimization problems." *IEEE Transactions on Evolutionary Computation*, 1(2), 129 - 140.

Kita, E., and Tanie, H. (1998). "GA-based topology optimization of continuum structures." Structural Optimization (Proceedings of the Australasian Conference on Structural Optimization), G. P. Steven, O. M. Querin, H. Guan, and Y. M. Xie, eds., Oxbridge Press, Victoria, 87-94.

Kita, E., and Tanie, H. (1999). "Topology and shape optimization of continuum structures using GA and BEM." *Structural Optimization*, 17(2/3), 130-139.

Kita, E., and Toyoda, T. (2000). "Structural design using cellular automata." *Structural and Multidisciplinary Optimization*, 19, 64-73.

Kita, E., and Toyoda, T. (2001). "Truss structural design based on concept of cellular automata." *Proceedings of the International Association for Shell and Spatial Structures Symposium (IASS2001)*, Nagoya, Japan.
Koestler, A. (1990). *The act of creation*, Arkana.
Kolmogorov, A. N. (1958). "A new invariant for transitive dynamical systems." *Doklady Akademii Nauk Souiza Sovetskikh Sotsialisticheskikh Respuplik*, 119, 861-864.
Koumousis, V. K., and Georgiou, P. G. (1994). "Genetic algorithms in discrete optimization of steel truss roofs." *Journal of Computing in Civil Engineering*, 8(3), 309-325.
Kowalczyk, R. (1997). "Constraint consistent genetic algorithms." *Proceedings of the Fourth IEEE International Conference on Evolutionary Computation (ICEC'97)*, Indianapolis, USA, 343 -348.
Koza, J. R. (1992). *Genetic programming : on the programming of computers by means of natural selection*, MIT Press, Cambridge, Mass.
Koziel, S., and Michalewicz, Z. (1999). "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization." *Evolutionary Computation*, 7(1), 19-44.
Kuhn, T. ([1962] 1970). *The structure of scientific revolutions*, University of Chicago Press, Chicago.
Kundu, S., Oda, J., and Koishi, T. (1997). "A self-organizing approach to optimization of structural plates using cellular automata." *Proceedings of the Second World Congress on Structural and Multidisciplinary Optimization (WCSMO-2)*, W. Gutkowski and Z. Mroz, eds., Zakopane, Poland, 173-180.
Lander, S. E. (1997). "Issues in multiagent design systems." *IEEE Expert*, 12(2), 18-26.
Lawo, M., and Thierauf, G. (1982). "Optimal design for dynamic stochastic loading: a solution by random search." Optimization in Structural Design, Bibl. Inst. Mannheim, University of Siegen, 346-352.
Le, T. V. (1995). "A fuzzy evolutionary approach to constrained optimization problems." *Proceedings of the Second IEEE International Conference on Evolutionary Computation (ICEC'95)*, Perth, Australia, 274-278.
Le, T. V. (1996). "A fuzzy evolutionary approach to constrained optimisation problems." *Proceedings of the Third IEEE International Conference on Evolutionary Computation (ICEC'96)*, T. Fukuda and T. Furuhashi, eds., Nagoya, Japan, 274-278.
Liepins, G. E., and Potter, W. D. (1991). "A genetic algorithm approach to multiple-fault diagnosis." Handbook of genetic Algorithms, L. Davis, ed., Van Nostrand Reinhold, New York, 237-250.
Liepins, G. E., and Vose, M. D. (1990). "Representational issues in genetic optimization." *Journal of Experimental and Theoretical Artificial Intelligence*, 2, 101-115.
Lin, C.-Y., and Hajela, P. (1993). "Genetic search strategies in large scale optimization." *Proceedings of the 34th AIAA/ASCE/ASME/AHS Structural Dynamics and Material Conference*, La Jolla, CA, 2437-2447.
Locke, J. ([1690] 1894). *An essay concerning human understanding*, Clarendon Press, Oxford.
Lorenz, E. N. (1963). "Deterministic nonperiodic flow." *Journal of Atmospheric Sciences*, 20, 130-141.
Lotka, A. J. (1925). *Elements of physical biology*, The Williams and Wilkins Co., Baltimore.

Luke, S. (2000). "Issues in scaling genetic programming: breeding strategies, tree generation, and code bloat," Ph.D. Dissertation, Department of Computer Science, University of Maryland, College Park, Maryland.

Luke, S., and Wiegand, R. P. (2002). "When coevolutionary algorithms exhibit evolutionary dynamics." *Proceedings of the Workshop on Understanding Coevolution: Theory and Analysis of Coevolutionary Algorithms (GECCO 2002)*, A. Barry, ed., New York, 236-241.

Maher, M. L. (1994). "Creative design using a genetic algorithm." *Computing in Civil Engineering*, 2014-2021.

Maher, M. L., and Poon, J. (1995). "Evolving a fitness landscape for design exploration." *Proceedings of the International Conference on Evolutionary Computation*, Perth, Australia.

Maher, M. L., and Poon, J. (1996). "Modelling design exploration as co-evolution." *Microcomputers in Civil Engineering*, 11(3), 195-210.

Maher, M. L., Poon, J., and Boulanger, S. (1996). "Formalising design exploration as co-evolution: a combined gene approach." Advances in Formal Design Methods for CAD, J. S. Gero and F. Sudweeks, eds., Chapman & Hall, 1-28.

Maher, M. L., and Wu, P. X. (1998). "Creativity through co-evolutionary design." Preprints of Computational Models of Creative Design, J. S. Gero, M. L. Maher, and F. Sudweeks, eds., Key Center of Design Computing, Sydney, Australia, 244-259.

Marchese, F. M. (2002). "A path-planner for mobile robots of generic shape with multilayered cellular automata." *Proceedings of the 5th International Conference on Cellular Automata for Research and Industry (ACRI 2002)*, S. Bandini, B. Chopard, and M. Tomassini, eds., Geneva, Switzerland, 178-189.

Marinosson, S. F., Chrobok, R., Pottmeier, A., Joachim, W., and Schreckenberg, M. (2002). "Simulation framework for the autobahn traffic in North Rhine-Westphalia." *Proceedings of the 5th International Conference on Cellular Automata for Research and Industry (ACRI 2002)*, S. Bandini, B. Chopard, and M. Tomassini, eds., Geneva, Switzerland, 315-324.

Maynard Smith, J. (1982). *Evolution and the theory of games*, Cambridge University Press.

Michalewicz, Z. (1995). "A survey of constraint handling techniques in evolutionary computation methods." *Proceedings of the 4th Annual Conference on Evolutionary Programming*, Cambridge, MA, 135-155.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*, 3rd rev. and extended edition, Springer-Verlag, Berlin ; New York.

Michalewicz, Z. (2000a). "Decoders." Evolutionary computation 2: advanced algorithms and operators, T. Bäck, D. B. Fogel, and Z. Michalewicz, eds., Institute of Physics Publishing, Bristol and Philadelphia, 49-55.

Michalewicz, Z. (2000b). "Repair algorithms." Evolutionary computation 2: advanced algorithms and operators, T. Bäck, D. B. Fogel, and Z. Michalewicz, eds., Institute of Physics Publishing, Bristol and Philadelphia, 56-61.

Michalewicz, Z., and Attia, N. F. (1994). "Evolutionary optimization of constrained problems." *Proceedings of the Third Annual Conference on Evolutionary Programming*, A. V. Sebald and L. J. Fogel, eds., San Diego, CA, USA, 98-108.

Michalewicz, Z., Dasgupta, D., Le Riche, R. G., and Schoenauer, M. (1996). "Evolutionary algorithms for constrained engineering problems." *Computers and Industrial Engineering Journal*, 30(4), 851-830.

Michalewicz, Z., and Nazhiyath, G. (1995). "Genocop III: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints." *Proceedings of the Second IEEE International Conference on Evolutionary Computation (ICEC'95)*, D. B. Fogel, ed., Perth, Australia, 647-651.

Michalewicz, Z., and Schoenauer, M. (1996). "Evolutionary algorithms for constrained parameter optimization problems." *Evolutionary Computation*, 4(1), 1-32.

Moore, E. F. (1962). "Machine models of self-replication." Mathematical Problems in the Biological Sciences (Proceedings of Symposia in Applied Mathematics), R. E. Bellman, ed., American Mathematical Society, Providence, RI, 17-33.

Mostow, J. (1985). "Toward better models of the design process." *AI Magazine*, 6(1), 44-57.

Mühlenbein, H. (1992). "Parallel genetic algorithms in combinatorial optimization: new developments in their interfaces." Computer Science and Operations Research, O. Balci, R. Sharda, and S. A. Zenios, eds., Pergamon Press, New York, 441-456.

Mühlenbein, H., and Schlierkamp-Voosen, D. (1993). "Predictive models for the breeder genetic algorithms. Continuous parameter optimization." *Evolutionary Computation*, 1(1), 25 - 49.

Mullins, S., and Rinderle, J. R. (1991). "Grammatical approaches to engineering design. Part 1: an introduction and commentary." *Research in Engineering Design*, 2(3), 121–135.

Murawski, K., Arciszewski, T., and De Jong, K. A. (2001). "Evolutionary computation in structural design." *Journal of Engineering with Computers*, 16, 275-286.

Mustafa, M., and Arciszewski, T. (1992). "Inductive learning of wind bracing design for tall buildings." Knowledge Acqusition in Civil Engineering, T. Arciszewski and L. Rossman, eds., American Society of Civil Engineers, New York, 190-203.

Myung, H., Kim, J.-H., and Fogel, D. B. (1995). "Preliminary investigation into a two-stage method of evolutionary optimization on constrained problems." *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, eds., Cambridge, MA, USA, 449-463.

Nagel, K. (2002). "Cellular automata models for transportation applications." *Proceedings of the 5th International Conference on Cellular Automata for Research and Industry (ACRI 2002)*, S. Bandini, B. Chopard, and M. Tomassini, eds., Geneva, Switzerland, 20-31.

Nair, P. B., and Keane, A. J. (2002). "Coevolutionary architecture for distributed optimization of complex coupled systems." *AIAA Journal*, 40(7), 1434-1443.

Nair, P. B., Keane, A. J., and Shimpi, R. P. (1998). "Combining approximation concepts with genetic algorithm-based structural optimization procedures." *Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Long Beach, CA, 1741-1751.

Nakanishi, Y., and Nakagiri, S. (1996). "Optimization of frame topology using boundary cycle and genetic algorithms." *JSME International Journal, Series A*, 39, 279-285.

Nakanishi, Y., and Nakagiri, S. (1997). "Structural optimization under topological constraint represented by homology groups." *JSME International Journal, Series A*, 40, 219-227.

Nakano, R., and Yamada, T. (1991). "Conventional genetic algorithm for job shop problems." *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA'91)*, R. K. Belew and L. B. Booker, eds., San Diego, CA, USA, 474-479.

Nanakorn, P., and Meesomklin, K. (2001). "An adaptive penalty function in genetic algorithms for structural design optimization." *Computers & Structures*, 79(29-30), 2527-2539.

Newsome, S. L., Spillers, W. R., and Finger, S. (1988). "Design theory '88." *Proceedings of the NSF Grantee Workshop on Design Theory and Methodology*, Troy, NY.

Nishino, F., and Duggal, R. (1990). "Shape optimum design of trusses under multiple loading." *Journal of Solids and Structures*, 19, 17-27.

Obayashi, S. (1998). "Pareto genetic algorithm for aerodynamic design using the Navier-Stokes equations." Genetic algorithms and evolution strategies in engineering and computer science: recent advances and industrial applications, D. Quagliarella, J. Periaux, C. Poloni, and G. Winter, eds., John Wiley & Sons, Chichester, England, 245-266.

Obayashi, S. (2002). "Pareto solutions of multipoint design of supersonic wings using evolutionary algorithms." Adaptive Computing in Design and Manufacture V, I. C. Parmee, ed., Springer-Verlag, London, 3-16.

Ohmori, H., and Kito, N. (1998). "Structural optimization of truss topology by genetic algorithms." *Journal of Theoretical and Applied Mechanics*, 47, 331-340.

Ohsaki, M. (1995). "Genetic algorithms for topology optimization of trusses." *Computers & Structures*, 57(2), 219-225.

Orvosh, D., and Davis, L. (1994). "Using a genetic algorithm to optimize problems with feasibility constraints." *Proceedings of the First IEEE International Conference on Evolutionary Computation (ICEC'94)*, Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, eds., Orlando, FL, USA, 548 -553.

Packard, N. H. (1988). "Adaptation toward the edge of chaos." Dynamic Patterns in Complex Systems, J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger, eds., World Scientific, Singapore, 293-301.

Pagie, L., and Mitchell, M. (2002). "A comparison of evolutionary and coevolutionary search." *International Journal of Computational Intelligence and Applications*, 2(1), 53-69.

Pahl, G., and Beitz, W. (1984). *Engineering design*, Springer-Verlag, Berlin.

Pahl, G., and Beitz, W. (1996). *Engineering design: a systematic approach*, Springer Verlag, New York.

Paredis, J. (1994). "Co-evolutionary constraints satisfaction." *Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN-III)*, Y. Davidor, H.-P. Schwefel, and R. Männer, eds., Jerusalem, Israel, 46-55.

Paredis, J. (1995). "Artificial coevolution, explorations in artifical life." AI Expert Presents, Miller Freeman Inc.

Pareto, V. (1896). *Cours D'Economie Politique*, F. Rouge, Lausanne.

Parmee, I. C. (1995). "Diverse evolutionary search for preliminary whole system design." *Proceedings of the 4th International Conference on AI in Civil and Structural Engineering*, Cambridge University.

Parmee, I. C. (1996). "The maintenance of search diversity for effective design space decomposition using cluster-oriented genetic algorithms (COGAs) and multi-agent strategies (GAANT)." *Proceedings of the Adaptive Computing in Engineering Design and Control*, University of Plymouth, UK.

Parmee, I. C. (1998a). "Adaptive computing in design and manufacture : the integration of evolutionary and adaptive computing technologies with product/system design and realisation." Springer-Verlag, London; New York.

Parmee, I. C. (1998b). "Genetic algorithms, and hydropower system design." *Computer-Aided Civil and Infrastructure Engineering*, 13(1), 31-41.

Parmee, I. C. (1999). "Exploring the design potential of evolutionary search, exploration and optimisation." Evolutionary Design by Computers, P. J. Bentley, ed., Academic Press Ltd., London.

Parmee, I. C. (2001). *Evolutionary and adaptive computing in engineering design*, Springer, London, New York.

Parmee, I. C. (2002). "Adaptive computing in design and manufacture V." Springer-Verlag, London, New York, xi, 392.

Parmee, I. C., and Purchase, G. (1994). "The development of a directed genetic search technique for heavily constrained design spaces." *Proceedings of the First International Conference on Adaptive Computing in Engineering Design and Control*, I. C. Parmee, ed., Plymouth, UK, 97-102.

Parmee, I. C., Vekeria, H. D., and Bilchev, G. (1997). "The role of evolutionary and adaptive search during whole system, constrained and detailed design optimization." *Engineering Optimization*, 29, 151-176.

Pedersen, K. (1999). "Designing platform families: an evolutionary approach to developing engineering systems," Ph.D. Dissertation, The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Pedersen, K., Emblemsvag, J., Bailey, R., Allen, J. K., and Mistree, F. (2000). "Validating design methods and research: the validation square." *Proceedings of the ASME Design Theory and Methodology Conference*, D. Thurston, ed., Baltimore, Maryland, DETC2000/DTM-14579.

Pedersen, P. (1987). "Optimal joint positions for space structures." *Journal of Structural Engineering*, 99(10), 2459-2477.

Peirce, C. S. (1998). *Charles S. Peirce: the essential writings*, Prometheus Books.

Periaux, J., and Winter, G. (1995). "Genetic algorithms in engineering and computer science." John Wiley, Chichester, UK.

Pezeshk, S. (2002). "State of the art on the use of genetic algorithms in design of steel structures." Recent Advances in Optimal Structural Design, S. Burns, ed., American Society of Civil Engineers, Reston, VA.

Pezeshk, S., Camp, C. V., and Chen, D. (2000). "Design of framed structures by genetic optimization." *Journal of Structural Engineering*, 126(3), 382-388.

Pironneau, O. (1984). *Optimal shape design for elliptic systems*, Springer-Verlag.

Poincaré, H. (1897). *Les Méthodes Nouvelles de la Méchanique Céleste*, Gauthier-Villars, Paris.

Poon, J., and Maher, M. L. (1996a). "Co-evolution and emergence in design." *Proceedings of the Workshop on Evolutionary Systems in Design AID'96*.

Poon, J., and Maher, M. L. (1996b). "Emergent behaviour in co-evolutionary design." Artificial Intelligence in Design '96, J. S. Gero, ed., Kluwer Academic Press.

Poon, J., and Maher, M. L. (1997). "Co-evolution and emergence in design." *Artificial Intelligence in Engineering*, 11(3), 319-327.

Popovici, E. (2003). "The bleeding edge of inventive design." George Mason University, Fairfax, VA.

Potter, M. A. (1997). "The design and analysis of a computational model of cooperative coevolution," Ph.D. dissertation, Computer Science Department, George Mason University, Fairfax, VA.

Potter, M. A., and De Jong, K. A. (1994). "A cooperative coevolutionary approach to function optimization." *Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN-III)*, Y. Davidor, H.-P. Schwefel, and R. Männer, eds., Jerusalem, Israel, 249-257.

Potter, M. A., and De Jong, K. A. (2000). "Cooperative coevolution: an architecture for evolving coadapted subcomponents." *Evolutionary Computation*, 8(1), 1-29.

Powell, D. J., and Skolnick, M. M. (1993). "Using genetic algorithms in engineering design optimization with non-linear constraints." *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93)*, S. Forrest, ed., Urbana-Champaign, IL, USA, 424-431.

Pullmann, T., Skolicki, Z., Freischlad, M., Arciszewski, T., De Jong, K. A., and Schnellenbach-Held, M. (2003). "Structural design of reinforced concrete tall buildings: evolutionary computation approach using fuzzy sets." *Proceedings of the 10th International Workshop of the European Group for Intelligent Computing in Engineering (EG-ICE)*, O. Ciftcioglu and E. Dado, eds., Delft, The Netherlands.

Rajan, S. D. (1995). "Sizing, shape, and topology design optimization of trusses using genetic algorithm." *Journal of Structural Engineering*, 121, 1480-1487.

Rajeev, S., and Krishnamoorthy, C. S. (1992). "Discrete optimization of structures using genetic algorithms." *Journal of Structural Engineering*, 118(5), 1233-1250.

Rajeev, S., and Krishnamoorthy, C. S. (1997). "Genetic algorithms-based methodologies for design optimization of trusses." *Journal of Structural Engineering*, 123(3), 350-358.

Ramasamy, J. V., and Rajasekaran, S. (1996). "Artificial neural network and genetic algorithm for the design optimizaton of industrial roofs – a comparison." *Computers & Structures*, 58(4), 747-755.

Rasheed, K. (1998). "An adaptive penalty approach for constrained genetic-algorithm optimization." *Proceedings of the Third Annual Genetic Programming Conference*, J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. L. Riolo, eds., Madison, Wisconsin, USA, 584-590.

Rechenberg, I. (1965). *Cybernetic solution path of an experimental problem*, Royal Aircraft Establishment, Farnborough, UK.

Rechenberg, I. (1973). *Evolutionsstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart-Bad Cannstatt.

Reynolds, R. G. (1994). "An introduction to cultural algorithms." *Proceedings of the Third Annual Conference on Evolutionary Programming*, A. V. Sebald and L. J. Fogel, eds., Singapore, 131-139.

Reynolds, R. G., Michalewicz, Z., and Cavaretta, M. J. (1995). "Using cultural algorithms for constraint handling in Genocop." *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, eds., Cambridge, MA, USA, 289-305.

Richards, R., and Sheppard, S. D. (1992). "Learning classifier systems in design optimization." *Proceedings of the 1992 Design Theory and Methodology Conference*, Scottsdale, Arizona, 179-186.

Richardson, J. T., Palmer, M. R., Liepins, G. E., and Hilliard, M. R. (1989). "Some guidelines for genetic algorithms with penalty functions." *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, J. D. Schaffer, ed., Fairfax, VA, USA, 191-197.

Rinderle, J. R. (1991). "Grammatical approaches to engineering design. Part II: Melding configuration and parametric design using attribute grammars." *Research in Engineering Design*, 2(3), 137–146.

Ritzel, B. J., Eheart, J. W., and Ranjithan, S. (1994). "Using genetic algorithms to solve a multiple objective groundwater pollution containment problem." *Water Resources Research*, 30(5), 1589-1603.

Rosenberg, R. S. (1967). "Simulation of genetic populations with biochemical properties," Ph.D. Dissertation, University of Michigan, Ann Harbor, Michigan.

Rosenman, M. (1997). "The generation of form using evolutionary approach." Evolutionary algorithms in engineering applications, D. Dasgupta and Z. Michalewicz, eds., Springer, Berlin New York, 69-86.

Rosenman, M., and Gero, J. S. (1999). "Evolving designs by generating useful complex gene structures." Evolutionary design by computers, P. J. Bentley, ed., Morgan Kaufmann Publishers, San Francisco, CA.

Rosin, C. D., and Belew, R. K. (1996). "New methods for competitive coevolution." *CS96-491*, Department of Computer Science and Engineering, University of California, San Diego, CA.

Roston, G. P. (1994). "A genetic methodology for configuration design," Ph.D. Dissertation, Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA.

Rothlauf, F. (2002). *Representations for genetic and evolutionary algorithms*, Physica-Verlag, Heidelberg New York.

Rowe, J. E. (2001). "The dynamical systems model of the simple genetic algorithm." Theoretical Aspects of Evolutionary Computing, L. Kallel, B. Naudts, and A. Rogers, eds., Springer Verlag, Berlin Heidelberg, 31-57.

Rozvany, G. I. N., Bendsoe, M. P., and Kirsch, U. (1995). "Layout optimization of structures." *Applied Mechanics Reviews*, 48(2), 41-120.

Russel, B. (1962). *Scientific outlook*, W.W. Norton & Company.

Sakamoto, J., and Oda, J. (1993). "Technique for optimal layout design for truss structures using genetic algorithms." *Proceedings of the 34th AIAA/ASCE/ASME/AHS Structural Dynamics and Material Conference AIAA/ASME Adaptive Structures Forum*, New York, NY, 2402-2408.

Sandgren, E., and Jensen, E. D. (1992). "Automotive structural design employing a genetic optimization algorithm." *Proceedings of the SAE International Congress and Exposition*, Detroit, Michigan, SAE Technical Paper #920772.

Sandgren, E., Jensen, E. D., and Welton, J. (1990). "Topological design of structural components using genetic optimization methods." Sensitivity Analysis and Optimization with Numerical Methods, AMD-vol. 115, Proceedings of the Winter Annual Meeting of the American Society of Mechanical Engineers, Dallas, TX, 31-43.

Sarma, K. C., and Adeli, H. (2001). "Bilevel parallel genetic algorithms for optimization of large steel structures." *Computer-Aided Civil and Infrastructure Engineering*, 16, 295-304.

Schaffer, J. D. (1984). "Some experiments in machine learning using vector evaluated genetic algorithms," Ph.D. Dissertation, Vanderbilt University, Nashville, TN.

Schaffer, J. D. (1985). "Multiple objective optimization with vector evaluated genetic algorithms." *Proceedings of the First International Conference on Genetic Algorithms (ICGA'85)*, J. J. Grefenstette, ed., Pittsburgh, PA, USA, 93-100.

Schaffer, J. D., and Eshelman, L. J. (1991). "On crossover as an evolutionarily viable strategy." *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA'91)*, R. K. Belew and L. B. Booker, eds., San Diego, CA, USA, 61-68.

Schmidt, L. C., and Cagan, J. (1994). "Recursive annealing: a computational model for machine design." *Research in Engineering Design*, 7(2), 102-125.

Schmidt, L. C., and Cagan, J. (1998). "Optimal configuration design: an integrated approach using grammars." *Journal of Mechanical Design*, 120(1), 2-9.

Schmit, L. A. (1981). "Structural synthesis- its genesis and development." *AAAI Journal*, 19(10), 1249-1263.

Schoenauer, M. (1996). "Shape representations and evolution schemes." *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, L. J. Fogel, P. J. Angeline, and T. Bäck, eds., San Diego, CA, USA.

Schoenauer, M., and Michalewicz, Z. (1996). "Evolutionary computation at the edge of feasibility." *Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature (PPSN-IV)*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, eds., Berlin, Germany, 245-254.

Schoenauer, M., and Michalewicz, Z. (1997). "Boundary operators for constrained parameter optimization problems." *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA'97)*, T. Bäck, ed., East Lansing, MI, USA, 320-329.

Schoenauer, M., and Wu, Z. (1993). "Discrete optimal design of structures by genetic algorithms." *Proceedings of the Conference Nationale sur le Calcul de Structures*, B. e. al., ed., Hermes, 833-842.

Schoenauer, M., and Xanthakis, S. (1993). "Constrained GA optimization." *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93)*, S. Forrest, ed., Urbana-Champaign, IL, USA, 573-580.

Schum, D. (2001). "IT944 lecture notes." George Mason University, Fairfax, VA.

Schwefel, H.-P. (1965). "Kybernetische Evolution als Strategie der experimentelen Forschung in der Stromungstechnik," Master's thesis, Hermann Föttinger Institute for Hydrodynamics, Technical University of Berlin.

Schwefel, H.-P. (1977). *Numerische Optimierung von Computer-modellen mittels der Evolutionsstrategie*, Birkhaeuser Verlag, Basel.

Schwefel, H.-P. (1981). *Numerical optimization of computer models*, John Wiley & Sons, Chichester, UK.

Sendhoff, B., Kreutz, M., and Seelen, W. v. (1997). "A condition for the genotype-phenotype mapping: causality." *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA'97)*, T. Bäck, ed., East Lansing, MI, USA, 354-361.

Shai, O. (2001). "Combinatorial representations in structural analysis." *Journal of Computing in Civil Engineering*, 15(3), 193-207.

Shankar, N., and Hajela, P. (1991). "Heuristics driven strategies for near-optimal structural topology development." Artificial Intelligence and Structural Engineering, B. H. V. Topping, ed., Civil-Comp Press, Oxford, UK, 219-226.

Shea, K., Cagan, J., and Fenves, S. J. (1997). "A shape annealing approach to optimal truss design with dynamic grouping of members." *Journal of Mechanical Design*, 119, 388-394.

Shen, W., and Barthes, J.-P. (1995). "DIDE: a multi-agent environment for engineering design." *Proceedings of the First International Conference on Multiagent Systems*, Menlo Park, CA, 344-351.

Shen, W., and Norrie, D. H. (1999). "Agent-based systems for intelligent manufacturing: state-of-the-art survey." *Knowledge and Information Systems*, 1(2), 129-156.

Shrestha, S. M., and Ghaboussi, J. (1998). "Evolution of optimum structural shapes using genetic algorithm." *Journal of Structural Engineering*, 124(11), 1331-1338.

Simon, H. A. (1981). *The sciences of the artificial*, MIT Press, Cambridge, MA.

Singer, J. (1995). "Mental processes and brain architecture: confronting the complex adaptive systems of human thought (an overview)." The Mind, The Brain, and Complex Adaptive Systems, H. Morowitz and J. Singer, eds., Addison-Wesley, 1-9.

Sipper, M. (1997). *Evolution of parallel cellular machines : the cellular programming approach*, Springer, New York.

Slater, J. G. (1988). "The philosophy of logical atomism and other essays, 1914-19. The collected papers of Bertrand Russel." Unwin Hyman.

Smith, A. E., and Tate, D. M. (1993). "Genetic optimization using a penalty function." *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93)*, S. Forrest, ed., Urbana-Champaign, IL, USA, 499-503.

Smith, R. E., Forrest, S., and Perelson, A. S. (1993). "Searching for diverse, cooperative populations with genetic algorithms." *Evolutionary Computation*, 1(2), 127-149.

Soh, C. K., and Yang, J. (1996). "Fuzzy controlled genetic algorithm search for shape optimization." *Journal of Computing in Civil Engineering*, 10(2), 143-150.

Soh, C. K., and Yang, J. (1998). "Optimal layout of bridge trusses by genetic algorithms." *Computer-Aided Civil and Infrastructure Engineering*, 13(4), 247-254.

Soh, C. K., and Yang, Y. (2001). "Genetic programming-based approach for structural optimization." *Journal of Computing in Civil Engineering*, 31, 31-37.

Sokolowski, J., and Zolesio, J.-P. (1992). *Introduction to shape optimization. Shape sensitivity analysis*, Springer-Verlag.

Spears, W. M. (2000). *Evolutionary algorithms: the role of mutation and recombination*, Springer, Berlin ; New York.

Srinivas, N., and Deb, K. (1994). "Multiobjective optimization using nondominated sorting in genetic algorithms." *Evolutionary Computation*, 2(3), 221-248.

Stauffer, L., Ullman, D. G., and Dietterich, T. G. (1987). "Protocol analysis of mechanical engineering design." *Proceedings of the International Conference on Engineering Design*, Boston, MA.

Stiny, G. (1980). "Introduction to shape and shape grammars." *Environment and Planning B: Planning and Design*, 7(3), 343-351.

Suh, N. P. (1990). *The principles of design*, Oxford University Press, New York.

Surry, P. D., and Radcliffe, N. J. (1997). "The COMOGA method: constrained optimisation by multi-objective genetic algorithms." *Control and Cybernetics*, 26(3).

Surry, P. D., Radcliffe, N. J., and Boyd, I. D. (1995). "A multi-objective approach to constrained optimization of gas supply networks: the COMOGA method." *Proceedings of the AISB-95 Workshop on Evolutionary Computing*, T. C. Fogarty, ed., Sheffield, UK, 166-180.

Syswerda, G., and Palmucci, J. (1991). "The application of genetic algorithms to resource scheduling." *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA'91)*, R. K. Belew and L. B. Booker, eds., San Diego, CA, USA, 502-508.

Taranath, B. S. (1998). *Steel, concrete and composite design of tall buildings*, 2nd edition, McGraw-Hill, New York.

Tate, D. M., and Smith, A. E. (1995). "A genetic approach to the quadratic assignment problem." *Computers and Operations Research*, 22(1), 73-78.

Thompson, D. A. W. (1942). *On growth and form*, Cambridge University Press, Cambridge.

Thompson, D. E. (1999). *Design analysis: mathematical modeling of nonlinear systems*, Cambridge University Press, Cambridge, UK.

Tong, C., and Sriram, D. (1992a). "Artificial Intelligence in engineering design. Volume 1: design representations and models of routine design." Academic Press, Boston, MA.

Tong, C., and Sriram, D. (1992b). "Artificial Intelligence in engineering design. Volume 2: models of innovative design, reasoning about physical systems, and reasoning about geometry." Academic Press, Boston, MA.

Tong, C., and Sriram, D. (1992c). "Artificial Intelligence in engineering design. Volume 3: knowledge acquisition, commercial applications and integrated environments." Academic Press, Boston, MA.

Topping, B. H. V. (1983). "Shape optimization of skeletal structures: a review." *Journal of Structural Engineering*, 109, 1933-1951.

Topping, B. H. V., and de Barros Leite, J. P. (1998). "Parallel genetic models for structural optimization." *Engineering Optimization*, 31(1), 65-99.

Turing, A. (1936). "On computable numbers with an application to the Entscheidungsproblem." *Proceedings of London Mathematical Society*, 2(42), 230-265.

Ulam, S. M. (1952). "Random processes and transformations." *Proceedings of the International Conference on Mathematics*, 264-275.

Ulam, S. M. (1974). *Stanislaw Ulam: sets, numbers, and universes; selected works*, MIT Press, Cambridge, Mass.,.

Van Veldhuizen, D. A., and Lamont, G. B. (1998). "Multi-onjective evolutionary algorithm research: a history and analysis." *TR-98-03*, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Vanderplaats, G. N. (1975). "Design of structures for optimum geometry." *TMX-62-462*, NASA.

Vekeria, H. D., and Parmee, I. C. (1996). "The use of a co-operative multi-level CHC GA for structual shape optimisation." *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany.

Volterra, V. (1926). "Variazioni e fluttuazioni del numero d'individui in specie animali conviventi." *Mem. R. Accad. Naz. dei Lincei. Ser. VI*, 2.

von Neumann, J. (1951). "The general and logical theory of automata." Cerebral Mechanisms in Behavior: The Hixon Symposium, L. A. Jeffress, ed., Wiley, 1-32.

von Neumann, J. (1966). "Theory of self-reproducing automata." A. W. Burks, ed., University of Illinois Press, Urbana.

Vose, M. D. (1990). "Formalizing genetic algorithms." *Proceedings of the IEEE Workshop on Genetic Algorithms, Neural Nets, and Simulated Annealing Applied to Problems in Signal and Image Processing*, Glasgow, UK.

Vose, M. D. (1997). "Logarithmic convergence of random heuristic search." *Evolutionary Computation*, 4(4), 395-404.

Vose, M. D. (1999a). "Random heuristic search." *Theoretical Computer Science*, 229(1-2), 103-142.

Vose, M. D. (1999b). *The simple genetic algorithm : foundations and theory*, MIT Press, Cambridge, Mass.

Waldron, M., and Waldron, K. (1991). "Mechanical design: theory and methodology." Springer-Verlag, New York.

Watabe, H., and Okino, N. (1993). "A study of genetic shape design." *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93)*, S. Forrest, ed., Urbana-Champaign, IL, USA, 445-451.

Whitley, L. D. (1989). "The GENITOR algorithm and selection pressure: why ranked-based allocation of reproductive trials is best." *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, J. D. Schaffer, ed., Fairfax, VA, USA, 239–255.

Wibowo, F. X. N., and Besari, M. S. (1998). "Genetic algorithms in shape optimization of oval axially symmetrical shells." Structural Optimization (Proceedings of the Australasian Conference on Structural Optimization), G. P. Steven, O. M. Querin, H. Guan, and Y. M. Xie, eds., Oxbridge Press, Victoria, 103-111.

Wiegand, R. P. (2003). "An analysis of cooperative coevolutionary algorithms," Ph.D. Dissertation, Department of Computer Science, George Mason University, Fairfax, VA, USA.

Wiegand, R. P., Liles, W. C., and De Jong, K. A. (2001). "An empirical analysis of collaboration methods in cooperative coevolutionary algorithms." *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, L. Spector and E. D. Goodman, eds., San Francisco, CA, 1235-1242.

Wittgenstein, L. ([1921] 1961). *Tractatus logico-philosophicus*, Routledge and Kegan Paul, London.

Wolfram, S. (1983). "Statistical mechanics of cellular automata." *Reviews of Modern Physics*, 55, 601-644.

Wolfram, S. (2002). *A new kind of science*, Wolfram Media, Champaign, IL.

Wolfram, S. (2003). *The Mathematica book*, 5th edition, Wolfram Media, Champaign, IL.

Woon, S. Y., Querin, O. M., and Steven, G. P. (2001). "Structural application of a shape optimization method based on a genetic algorithm." *Structural and Multidisciplinary Optimization*, 22(1), 57–64.

Wright, A., and Agapie, A. (2001). "Cyclic and chaotic behavior in genetic algorithms." *Proceedings of the GECCO 2001: Genetic and Evolutionary Computation Conference*, L. Spector and E. D. Goodman, eds., San Francisco, CA, 718-724.

Wright, A., and Bidwell, G. L. (1997). "A search for counterexamples to two conjectures on the simple genetic algorithm." *Proceedings of the Fourth Workshop on Foundations of Genetic Algorithms*, R. K. Belew and M. D. Vose, eds., San Diego, CA, USA, 73-84.

Xie, Y. M., and Steven, G. P. (1992). "Shape and layout optimization via an evolutionary procedure." *Proceedings of the International Conference on Computational Engineering Science*, Hong Kong University of Science and Technology, Hong Kong.

Xie, Y. M., and Steven, G. P. (1997). *Evolutionary structural optimization*, Springer-Verlag, Berlin Heidelberg New York.

Yang, J., and Soh, C. K. (1997). "Structural optimization by genetic algorithms with tournament selection." *Journal of Computing in Civil Engineering*, 11(3), 195-200.

Yang, X., and Gen, M. (1994). "Evolution program for bicriteria transportation problem." *Proceedings of the 16th International Conference on Computers and Industrial Engineering*, M. Gen and T. Kobayashi, eds., Ashikaga, Japan, 451-454.

Yang, Y., and Soh, C. K. (2002). "Automated optimum design of structures using genetic programming." *Computers & Structures*, 80(18-19), 1537-1546.

Yoo, J., and Hajela, P. (1999). "Immune network simulations in multicriterion design." *Structural and Multidisciplinary Optimization*, 18(2-3), 85-94.

Zitzler, E., and Thiele, L. (1998). "An evolutionary algorithm for multi-objective optimization: the strength Pareto approach." *Technical Report 43*, Computer Engineering and Communication Networks Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland.

CURRICULUM VITAE

Rafal P. Kicinger earned a Master of Science degree in Civil Engineering from Warsaw University of Technology in 1999 and a Doctor of Philosophy degree from George Mason University in the fall of 2004. His interdisciplinary research interests include innovative approaches to engineering design inspired by the fundamental processes of biological development and evolution, cellular automata, and dynamical systems and chaos theory.

Rafal Kicinger co-authored 16 refereed papers, majority of them as first author. He made 10 presentations at major international conferences in Europe and in the United States. He is a recipient of several prestigious awards, including the Outstanding Graduate Student Award from the School of Information Technology and Engineering and the Outstanding Graduate Award from the Department of Civil, Environmental and Infrastructure Engineering, both at George Mason University. As a part of his research, he developed a unique research and design support tool called Emergent Designer which earned the Best Use of Technology Award during the Innovations 2004 Exhibition. He also received a merit scholarship from the Polish Ministry of Education in 1999. He is a Summa Cum Laude Graduate of the Department of Civil Engineering at Warsaw University of Technology.

Rafal Kicinger has served as a reviewer for Journal of Computing in Civil Engineering since 2002. He also reviewed papers submitted to the Special Session on Evolutionary Design at the Congress of Evolutionary Computation in 2004. He acted as a student Vice-President of the local chapter of Phi Beta Delta, Honor Society for International Scholars, George Mason University. He is an alumnus of the 1st New Kind of Science Summer School held at Brown University, Providence, RI.